



Escola Politècnica Superior  
de Castelldefels

UNIVERSITAT POLITÈCNICA DE CATALUNYA

# TREBALL DE FI DE CARRERA

**TÍTOL DEL TFC:** Diseño y prototipado de un sistema de monitorización  
de configuración de un UAV

**TITULACIÓ:** Enginyeria Tècnica Aeronàutica, especialitat Aeronavegació

**AUTORS:** Eloy Herrera Cañadas  
Víctor de los Santos Bernad

**DIRECTOR:** Jorge Ramírez Alcántara

**DATA:** 27 de Juliol de 2009

**Título:** Diseño y prototipado de un sistema de monitorización de configuración de un UAV

**Autores:** Eloy Herrera Cañadas  
Víctor de los Santos Bernad

**Director:** Jorge Ramírez Alcántara

**Fecha:** 27 de Julio de 2009

## **Resumen**

Este documento detalla la elaboración de un sistema de monitorización de la configuración de un vehículo aéreo no tripulado (UAV). La idea es crear un sistema que monitorizará en tiempo real la posición de las superficies de vuelo, permitiendo al responsable del UAV conocer mejor el estado del vehículo. Se ha realizado un proyecto intentando seguir las pautas reales en el mundo laboral para la realización de un proyecto aeronáutico, centrando la atención en la elaboración de un diseño y prototipado real del sistema a utilizar y en la realización de un estudio de seguridad y cómo afecta éste a la elaboración del diseño.

El diseño de un sistema electrónico es un proceso más o menos sencillo, dependiendo del tipo de sistema desarrollado. Sin embargo, en el mundo aeronáutico existen ciertas normativas que obligan a desarrollar los sistemas siguiendo unas pautas muy concretas, y que se deben seguir para poder llegar a certificar el sistema para su uso aeronáutico. Por ello se decide elaborar un proyecto siguiendo tales pautas.

En el documento quedará reflejada la elaboración de los diferentes pasos, explicando primero la idea de proyecto original que se tenía y pasando por los diferentes procesos que han llevado al diseño y prototipo final. Se tratarán aspectos como el diseño arquitectural y detallado del sistema, cómo se han elaborado los diferentes documentos de seguridad y cómo han afectado en el desarrollo del sistema, y los pasos realizados para la verificación del sistema. Además, se incluye un manual del sistema con el cual se puede adaptar fácilmente el sistema desarrollado a cualquier superficie que se desee monitorizar.

El documento incluye los diferentes diseños descartados, independientemente del estado de desarrollo al que se llegó, y las causas por las que se descartaron, poniendo especial atención en todas aquellas causas de seguridad que recomendaban cambiar el diseño.

El documento englobará la realización de todos los aspectos mencionados e incluirá todos los anexos en los que se muestra detalladamente la realización de cada proceso.

**Title:** UAV configuration monitoring system design and prototype

**Authors:** Eloy Herrera Cañadas  
Víctor de los Santos Bernad

**Director:** Jorge Ramírez Alcántara

**Date:** July, 27th 2009

## Overview

This document details the elaboration of an Unmanned Aerial Vehicle (UAV) configuration monitoring system. The idea is to create a system which will monitor in real time the position of a control surface, allowing the UAV manager to have more information about the system state. The project has been done trying to follow the real guidelines for the development of an aeronautical project, focusing the attention on the design and real prototype of the system to be used and the development of a Safety Assessment and how it affects the making of the design.

The design of an electronic system is a more or less complex process, depending on the kind of developed system. However, in an aeronautical environment specific standards which must be followed in order to certify systems for aeronautical purposes exist. That is the reason why a project following those guidelines is decided to be done.

This document reflects the making of the different steps, explaining first the original project which was proposed and going through the different processes that have driven to the final design and prototype. Aspects like architectural and detailed design, the elaboration of the safety documents and how they have affected the development of the system, and the steps done for the system verification would be treated. Also, a system manual is included for adapting the developed system easily to any surface that wants to be controlled.

The document includes the discarded designs, independently of the development state which was reached, and the causes which made them were discarded, focusing on the safety causes that recommended the change of the design.

The document will include all the mentioned aspects and will have attached all the annexes which show in detail the making of each process.

Dedicado a Jorge Ramírez, por su paciencia, confianza depositada y constante atención; a Óscar Casas y Josué Martínez, por su inestimable ayuda sin la cual esto no podría haber sido posible; a nuestros seres queridos (ellos saben quienes son), por el apoyo incondicional y darnos la fuerza necesaria para llevar a cabo el proyecto; y a la comunidad de profesores de la EPSC que en algún momento nos han ofrecido su ayuda cuando la hemos necesitado.



# ÍNDICE

<b>INTRODUCCIÓN .....</b>	<b>1</b>
<b>CAPÍTULO 1. PREPARACIÓN DEL PROYECTO .....</b>	<b>2</b>
1.1 Idea del proyecto .....	2
1.2 Ideas descartadas .....	3
1.2.1 Idea original .....	3
1.2.2 Sensores del motor .....	4
1.2.3 Transmisión de datos y monitorización en tierra .....	5
<b>CAPÍTULO 2. PLANIFICACIÓN.....</b>	<b>6</b>
2.1 Introducción .....	6
2.2 Recursos humanos .....	6
2.3 Herramientas utilizadas .....	7
2.4 Elaboración de <i>templates</i> para los documentos.....	8
2.5 Trazabilidad de requisitos y procesos .....	8
2.6 Gestión de configuración .....	10
2.6.1 Gestión de requisitos.....	10
2.6.2 Gestión de documentos.....	10
2.6.3 Lista de requisitos.....	10
2.6.4 Tareas y planificación.....	11
2.7 Presupuesto .....	12
<b>CAPÍTULO 3. FHA .....</b>	<b>13</b>
3.1 Introducción.....	13
3.1.1 FHA Nivel Avión: .....	13
3.1.2 FHA Nivel sistema .....	14
<b>CAPÍTULO 4. DISEÑO ARQUITECTURAL .....</b>	<b>20</b>
4.1 Introducción .....	20
4.2 Descripción del UAV .....	20
4.2.1 Información general .....	20
4.2.2 Superficies de control del UAV .....	21
4.2.3 Sistema de alimentación .....	23
4.2.4 Sistema de comunicaciones.....	23
4.3 Diseño De La Arquitectura .....	23
4.3.1 Decisiones iniciales .....	23
4.3.2 Resolución del sistema.....	24
4.3.3 Sistema de Alimentación .....	25

4.4	Consideraciones de seguridad durante el diseño arquitectural .....	26
<b>CAPÍTULO 5. PSSA .....</b>		<b>27</b>
5.1	Introducción .....	27
5.2	Lista de requisitos de safety del avión .....	28
5.3	Evaluar las decisiones de diseño con respecto a los requisitos y objetivos del sistema .....	29
5.4	Derivar los requisitos de “safety” al diseño de los subsistemas .....	31
5.4.1	Lista de condiciones de fallo actualizada que incluya unas directrices de cómo los requisitos pueden ser satisfechos. ....	31
5.4.2	Lista con los requisitos de “safety” interrelacionados con el hardware y software. ....	32
5.4.3	Requisitos para la instalación del sistema .....	32
5.4.4	DAL del software y el hardware.....	33
5.4.5	Tareas de mantenimiento.....	33
5.5	Trazabilidad de los requisitos derivados del PSSA.....	33
<b>CAPÍTULO 6. DISEÑOS DESCARTADOS .....</b>		<b>35</b>
6.1	Introducción .....	35
6.2	Sensores de Desplazamiento Angular: Potenciómetro .....	37
6.2.1	Tecnología del sensor .....	37
6.2.2	Motivos del descarte.....	38
6.3	Sensores capacitivos.....	40
6.3.1	Tecnología del sensor .....	40
6.3.2	Motivos del descarte.....	42
6.4	DECISIÓN FINAL: POTENCIÓMETRO DEL SERVO.....	42
<b>CAPÍTULO 7. ZSA.....</b>		<b>44</b>
7.1	Introducción .....	44
7.2	Localización de nuestro sistema dentro de la aeronave.....	45
7.2.1	Alerones.....	46
7.2.2	Timón de dirección .....	46
7.2.3	Timón de profundidad.....	47
7.3	Identificación de todos los subsistemas que sean un problema en potencia.....	48
7.3.1	Potenciómetro .....	48
7.3.2	Encapsulado del servomotor .....	49
7.3.3	Encapsulado de nuestro sistema (no puesto en práctica) .....	49
7.3.4	Malfuncionamiento general del sistema .....	49
7.4	Como evitaremos los anteriores problemas potenciales. ....	50
7.4.1	Potenciómetro .....	50
7.4.2	Encapsulado del servomotor .....	50
7.4.3	Encapsulado de nuestro sistema (no implementado) .....	50
7.4.4	Malfuncionamiento general del sistema .....	51

<b>CAPÍTULO 8. DISEÑO DETALLADO .....</b>	<b>52</b>
8.1 Introducción .....	52
8.2 Diseño del Hardware .....	53
8.3 Diseño del Software .....	55
8.4 Interfaz .....	57
8.5 Consideraciones de seguridad durante el diseño detallado .....	57
<b>CAPÍTULO 9. VERIFICACIÓN .....</b>	<b>59</b>
9.1 Introducción .....	59
9.2 Matriz de verificación .....	60
9.3 Reporte de problemas.....	61
9.4 Recopilación de documentos concernientes a la verificación .....	62
9.5 Trazabilidad.....	63
<b>CAPÍTULO 10. IMPLEMENTACIÓN .....</b>	<b>64</b>
10.1 Introducción .....	64
10.2 Resultado del hardware .....	65
10.3 Resultados del software .....	66
10.4 Documentación.....	67
10.4.1 Manual de instalación .....	67
10.4.2 Manual de calibración.....	68
10.4.3 Manual de usuario .....	68
<b>CAPÍTULO 11. CONCLUSIONES .....</b>	<b>69</b>
<b>CAPÍTULO 12. GLOSARIO.....</b>	<b>70</b>
<b>CAPÍTULO 13. BIBLIOGRAFÍA .....</b>	<b>71</b>

# ÍNDICE DE FIGURAS

Figura 1.1: Esquema del UAS .....	2
Figura 1.2: Desarrollo de Seguridad y Diseño en Paralelo.....	3
Figura 1.3: Esquema de funcionamiento de la idea original.....	4
Figura 1.4: Funciones incluidos y excluidos del desarrollo.....	5
Figura 2.1: Proceso de adopción del nuevo formato. ....	7
Figura 2.2: Esquema de los procesos de un proyecto. ....	9
Figura 2.3: Lista de Tareas.....	11
Figura 2.4: Diagrama de GANTT de la planificación inicial .....	12
Figura 2.5: Diagrama de GANTT de la planificación final.....	12
Figura 3.1: Ejemplo de condiciones de fallo para una función .....	13
Figura 3.2: Funciones de alto nivel de la aeronave .....	14
Figura 3.3: Funciones de nuestro sistema .....	15
Figura 3.4: Condiciones de fallo puntuales. ....	16
Figura 3.5: Efectos de las condiciones de fallo. ....	17
Figura 3.6: Criterios de aceptación.....	18
Figura 3.7: Trazabilidad entre procesos .....	19
Figura 4.1: Fotografía del modelo SHADOW-MK1 [1].....	21
Figura 4.2: Variación de la posición del servo en función de la duración del pulso.....	22
Figura 4.3: Conexión entre el servo y la superficie de control .....	22
Figura 4.4: Esquema del sistema .....	24
Figura 4.5: Diagrama de bloques de los subsistemas.....	24
Figura 4.6: Distribución de la alimentación.....	25
Figura 5.1: Condiciones de fallo para el alerón izquierdo.....	29
Figura 5.2: Diagrama de pérdida del sistema.....	29
Figura 5.3: DAL de los diferentes sistemas.....	30
Figura 5.4: Flujos de safety .....	34
Figura 6.1: Proceso iterativo para la realización del diseño .....	36
Figura 6.2: Proceso de diseño iterativo hasta el diseño final .....	37
Figura 6.3: Acondicionamiento del potenciómetro.....	38
Figura 6.4: Proceso iterativo realizado con el sensor resistivo .....	39
Figura 6.5: Adaptación de un sensor capacitivo.....	40
Figura 6.6: Ejemplo de la distancia a medir por un sensor capacitivo .....	41
Figura 6.7: Dibujo de localización del sensor .....	41
Figura 6.8: Circuito de control de lazo cerrado del servo .....	43
Figura 7.1: Ejemplo división del avión en subsistemas. ....	45
Figura 7.2: Ubicación de nuestro sistema y cableado de control .....	45
Figura 7.3: Servomotor del alerón izquierdo .....	46
Figura 7.4: Cavidad en la que instalar nuestro sistema.....	47
Figura 7.5: Servomotores del timón de profundidad.....	47
Figura 7.6: Funciones de alto nivel de un UAS .....	48
Figura 7.7: Cableado a extraer del servo. ....	50
Figura 8.1: División del diseño detallado y procedencia.....	52
Figura 8.2: Módulo MSP430 RF2500 con el microcontrolador MSP430F2274 .....	53
Figura 8.3: Módulo de Ethernet WIZ810MJ.....	54
Figura 8.4: Diseño del sistema sensor .....	54
Figura 8.5: Diseño de la PCB del sistema.....	55
Figura 8.6: División en funciones del software .....	56

Figura 8.7: División del software en ficheros.....	56
Figura 8.8: Interfaz del sistema .....	57
Figura 9.1: Proceso de verificación .....	60
Figura 10.1: Diagrama de implementación del sistema.....	64
Figura 10.2: Esquema de implementación del Hardware.....	65
Figura 10.3: Imagen de la placa del prototipo .....	65
Figura 10.4: PCB sin los módulos insertados.....	66
Figura 10.5: Esquema de implementación del software.....	66
Figura 10.6: Recepción de los datos del sistema.....	67

## ÍNDICE DE TABLAS

Tabla 2.1: Recursos Humanos .....	6
Tabla 2.2: Colaboradores .....	6
Tabla 2.3: Ejemplo de requisito .....	10
Tabla 3.1: Descripción del entorno y la lista de configuraciones de emergencia. .....	15
Tabla 4.1: Resolución de los diferentes subsistemas.....	25
Tabla 5.1: Extracto de tabla de requisitos derivados del PSSA.....	28
Tabla 5.2: Clasificación de las condiciones de fallo según su DAL .....	30
Tabla 5.3: Condiciones de fallo actualizadas .....	31
Tabla 5.4: Requisitos de “safety” interrelacionados con el hardware y software .....	32
Tabla 5.5: Requisitos para la instalación del sistema.....	32
Tabla 5.6: DAL de Software y hardware.....	33
Tabla 9.1: Extracto de la matriz de verificación .....	61
Tabla 9.2: Ejemplo de reporte de problema .....	62

# INTRODUCCIÓN

Actualmente en el sector de la aeronáutica está creciendo muy rápidamente el interés por los UAS, es decir, los sistemas aéreos no tripulados. Estos sistemas están compuestos principal y generalmente por un vehículo aéreo no tripulado, también conocido como UAV (siglas del inglés de *Unmanned Aerial Vehicle*) y un sistema de control en tierra, que monitoriza y realiza el seguimiento del UAV. La misma Universidad dispone de un departamento destinado a la investigación en los UAS, el grupo Ícarus, con un UAV para el cual hay varios equipos desarrollando diferentes sistemas.

La presencia de este grupo y el interés por el desarrollo de sistemas aviónicos llevó a la idea de realizar un proyecto de telemetría y monitorización de las superficies de vuelo del UAV disponible. Este sistema será capaz de medir mediante un sistema de sensores la posición de cada superficie del UAV (alerones, timón de profundidad y de dirección). Además, en la búsqueda de realizar un proyecto interesante, y alejarse de un simple desarrollo electrónico, se decidió realizar un estudio de seguridad para estudiar el diseño de un elemento aeronáutico desde ese punto de vista, viendo como afectaban al diseño la elaboración de los diferentes informes de seguridad.

Este documento muestra como se ha llevado, a pequeña escala, un proyecto aeronáutico y la importancia de la relación entre los procesos de seguridad y de diseño del sistema.

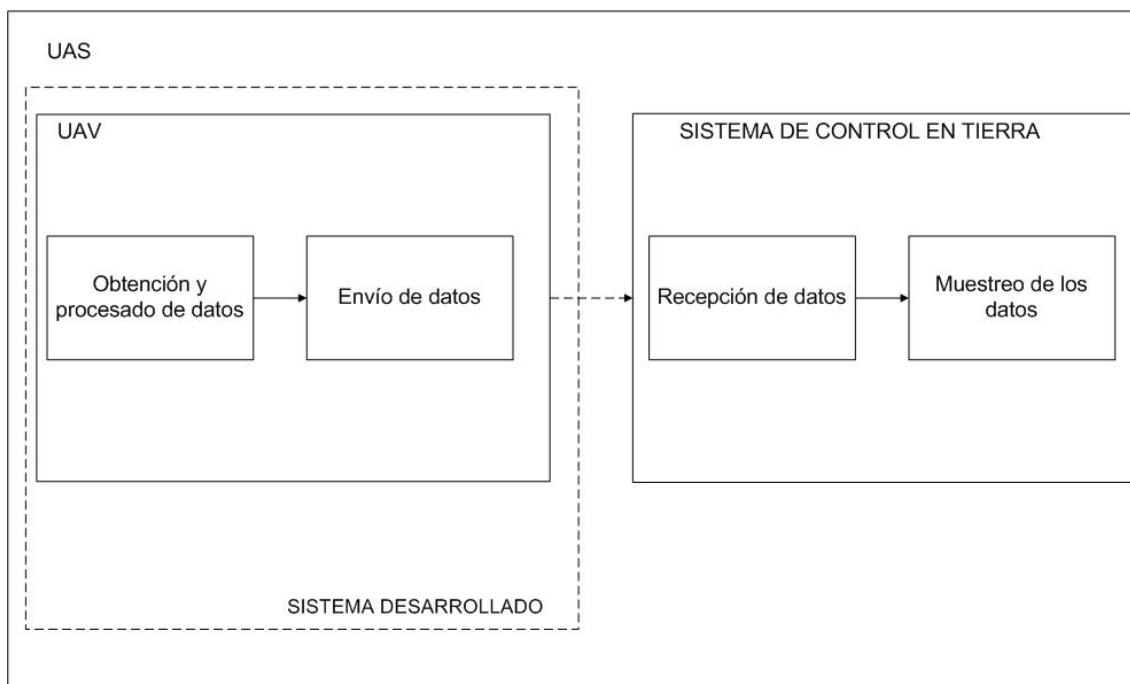
El documento se estructura en todas las fases de diseño y de seguridad realizadas, de forma cronológica con la forma en que se desarrolló el sistema. Cada parte detallará la metodología utilizada en ese apartado en concreto y las conclusiones extraídas de él.

Es importante discernir que a lo largo del trabajo se realizarán continuas alusiones a los documentos de seguridad. Estos documentos son los que en inglés se conocen como documentos de *Safety*. En español, *Safety* y *Security* se traducen como seguridad, sin embargo tienen significados diferentes. En inglés, *Security* es todo aquello que trata con evitar un mal uso del sistema. Sin embargo, *Safety* realiza un análisis del sistema para comprobar los problemas que pueden afectar al avión y provocar un daño en el avión o incluso en personas físicas (accidentes, etc.). Por tanto, siempre que en el documento se haga alusiones a la seguridad, se habla de *Safety* y no *Security*.

# CAPÍTULO 1. PREPARACIÓN DEL PROYECTO

## 1.1 Idea del proyecto

El proyecto consiste en la realización de un sistema de monitorización de superficies (monitorización de la configuración) de un UAV. Se desea conocer la posición en tiempo real de las superficies que controlan el UAV. Para ello, es necesario crear un sistema capaz de obtener, mediante el uso de sensores, la posición de las diferentes superficies. Tras esta obtención, el sistema debe emitir esa posición de forma que pueda ser obtenida por el responsable del sistema y conocer así las posiciones de las superficies, de forma que tenga más información del estado del avión en cada momento. El proyecto se centrará en la creación de un sistema que emita la posición, y cuya información se transmita de forma que sea fácilmente obtenible a través del sistema de comunicaciones del UAV. El UAV está dentro de un sistema UAS (Sistema Aéreo no tripulado), que contiene además el sistema de control en tierra. Este sistema de control en tierra es el que debe obtener y monitorizar los datos. Sin embargo, el proyecto se limitará a crear el sistema embarcado en el UAV. La **Figura 1.1** muestra un esquema del UAS y la localización del proyecto en él.

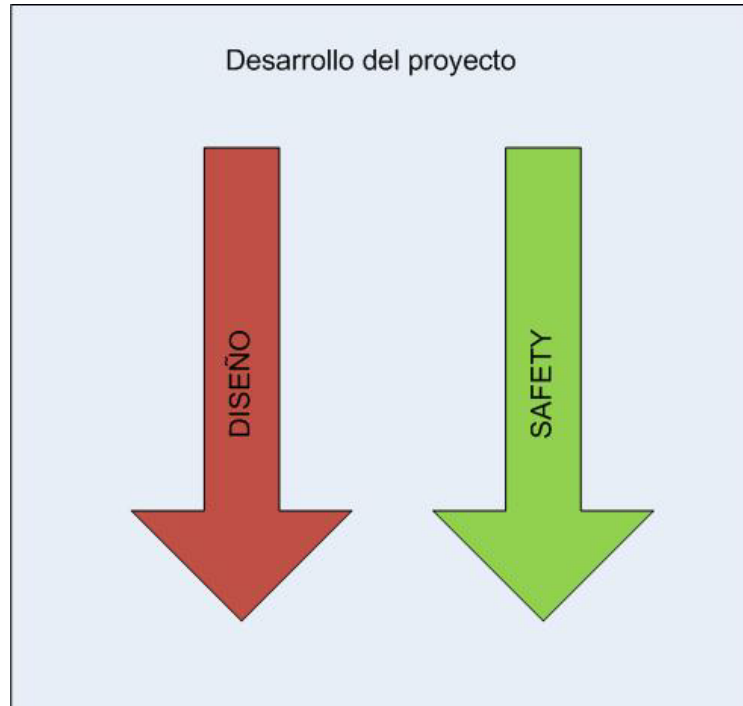


**Figura 1.1: Esquema del UAS**

El proyecto no solo incluye el diseño del sistema, sino un estudio detallado de cómo afecta el sistema al avión. Este estudio es un estudio de la seguridad del avión. En el mundo de la aeronáutica, cualquier modificación, por mínima que sea, en un vehículo aéreo supone un estudio de la forma en que puede variar



el correcto funcionamiento del sistema. El hecho de realizar este estudio ayudará a comprender mejor la toma de las diferentes decisiones del diseño, y como aun siendo un sistema no demasiado complejo, puede ver comprometido y pasar a ser crítico para el funcionamiento del avión. Este estudio de la seguridad debe realizarse en paralelo con el desarrollo del diseño del prototipo, tal como muestra la **Figura 1.2**.



**Figura 1.2:** Desarrollo de Seguridad y Diseño en Paralelo

## 1.2 Ideas descartadas

El desarrollo del proyecto pasó por diferentes fases y desde su inicio hubo muchos cambios importantes que llevaron a necesitar replantear partes importantes del proyecto, e incluso llegó a cambiar la idea original del proyecto de forma bastante notable como para poder decir que se trataban de proyectos totalmente diferentes.

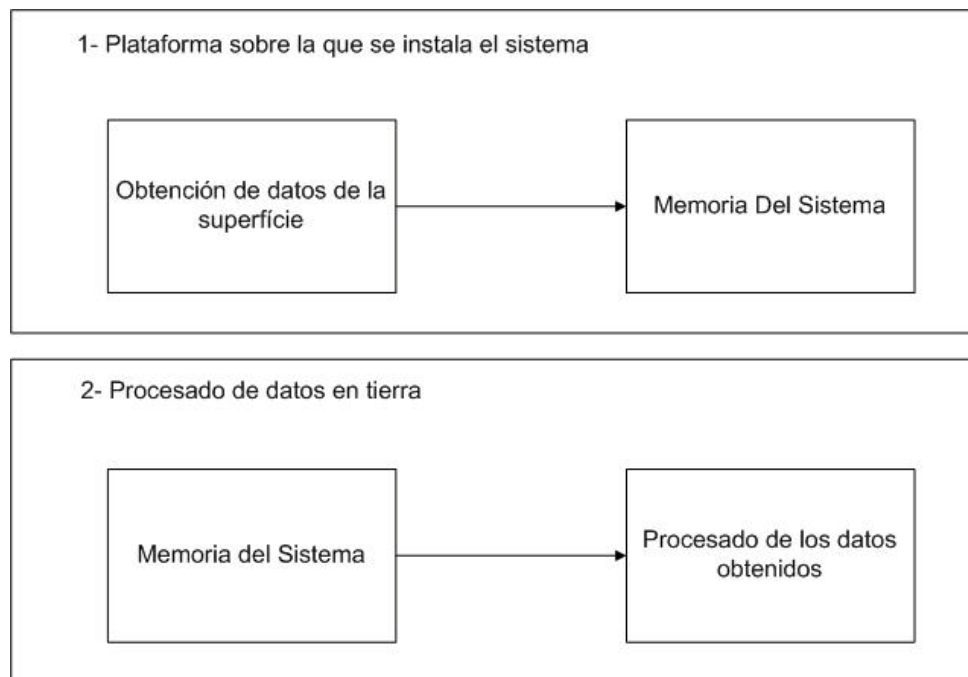
### 1.2.1 Idea original

El proyecto inicial consistía en realizar un sistema de monitorización para superficies. No se había pensado realizar un sistema pensado para el UAV del proyecto Ícarus.

El problema principal consistía en tener una plataforma sobre la que trabajar. Se pensó utilizar un avión o un helicóptero de aeromodelismo. También se planteó el uso de un coche de radio control, del cual se habrían monitorizado

aspectos como la potencia del motor o la dirección. El sistema habría consistido en un sistema sensor para las superficies a controlar. Sin embargo, una de las diferencias principales con lo que sería el proyecto final, reside en que el sistema no transmitiría la información en tiempo real, sino que guardaría la información en un módulo de memoria, al que se podría acceder cuando el vehículo estuviera parado. Además, se controlaría tanto la posición real de la superficie con la posición que debería tener, es decir, la orden mandada. Sería algo parecido a un sistema de cajas negra de los aviones.

El proyecto no era excesivamente atractivo debido a que no tenía mucho uso un sistema que no pudiera monitorizar en directo la información. Solo serviría en caso de que hubiera habido algún problema y se deseara obtener más información sobre lo sucedido, o si se deseara realizar una comprobación de parámetros. No era un sistema porque su utilidad era más bien escasa. De ahí que se decidiera acudir al grupo Icarus, con una disponibilidad de plataformas utilizables mayor, incluyendo un UAV. La **Figura 1.3** muestra un esquema de lo expuesto.



**Figura 1.3:** Esquema de funcionamiento de la idea original

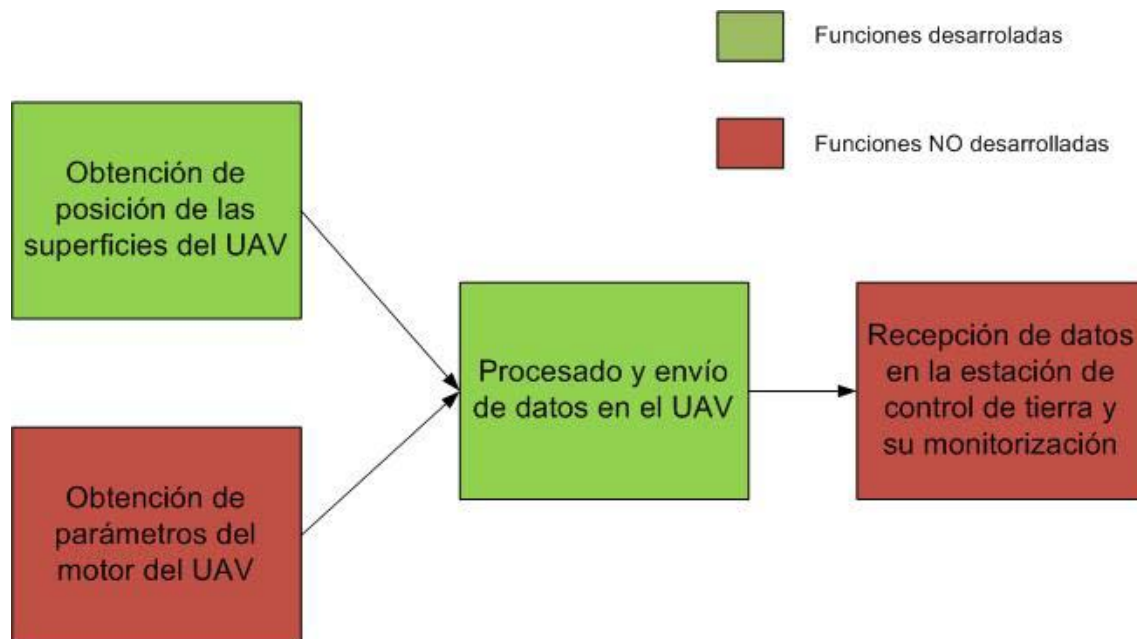
### 1.2.2 Sensores del motor

Cuando se ideaba el proyecto, además de querer controlar la posición de las diferentes superficies, se quería controlar parámetros tales como la temperatura, potencia o las revoluciones del motor. Sin embargo, de forma paralela a nuestro proyecto empezó un proyecto centrado exclusivamente en la monitorización de los parámetros del motor. El proyecto, titulado *Engine Monitoring for Unmanned Aerial Vehicles* y desarrollado por Julio Sagardoy,

estaba totalmente centrado en los aspectos de motor y sería un gasto de tiempo y recursos realizar dos sistemas para la misma función. En el momento que se conoció la realización de tal proyecto, se decidió dejar de lado este aspecto del diseño. Llegó a tener una serie de requisitos y una parte de diseño realizada antes de su descarte.

### 1.2.3 Transmisión de datos y monitorización en tierra

El proyecto también quería en un principio abarcar todo el proceso desde la obtención de los datos de los sensores al muestreo en tierra de los datos. La cancelación de la realización de este sistema se debe a que el UAV dispone de un sistema de comunicaciones que ya permite el envío de los datos a través de un radioenlace con tierra, y por tanto no se veía necesario el desarrollo de un sistema propio, pues sería redundante. El sistema de comunicaciones, llamado MAREA, se encuentra en desarrollo debido a que no está implementado en la arquitectura definitiva para el UAV. Para más información acerca de este sistema, revisar el apartado 4.2.4. El proyecto desarrollado aquí se dejó totalmente listo para su fácil adaptación al sistema de comunicaciones cuando fuera necesario. La **Figura 1.4** muestra un diagrama de los procesos incluidos y excluidos del proyecto final.



**Figura 1.4:** Funciones incluidos y excluidos del desarrollo

## CAPÍTULO 2. PLANIFICACIÓN

### 2.1 Introducción

Al abordar un proyecto aeronáutico debemos realizar una planificación de las tareas y procesos a realizar y definir una metodología a seguir. Estos proyectos suelen estar divididos entre multitud de procesos debido a su extensión. Esta división requiere una trazabilidad e interconexión entre los diferentes procesos, debiéndose así definir un protocolo de comunicación entre los diferentes departamentos.

### 2.2 Recursos humanos

En todo proyecto aeronáutico es importante describir todos los equipos que conforman el capital humano del proyecto. Esto es útil tanto para que cada empleado conozca las competencias que tiene dentro del proyecto como para que sepa a que departamento se debe dirigir en caso de duda.

En nuestro proyecto no fue difícil concretar el equipo humano con el que contábamos debido a lo reducido de la plantilla. En la **Tabla 2.1** podemos observar los recursos humanos adscritos al proyecto, mientras que en la **Tabla 2.1** se recoge todo aquel personal que nos ha brindado su experiencia y nos ha asistido en el transcurso del proyecto.

**Tabla 2.1:** Recursos Humanos

Personal	Función
Víctor de los Santos	Encargado del departamento de diseño.
Eloy Herrera	Encargado del departamento de seguridad.

**Tabla 2.2:** Colaboradores

Personal	Función
Jorge Ramírez	Supervisor de documentación.
Óscar Casas	Asistencia técnica. Sistema sensor.
Josué Martínez	Asistencia técnica. UAV.

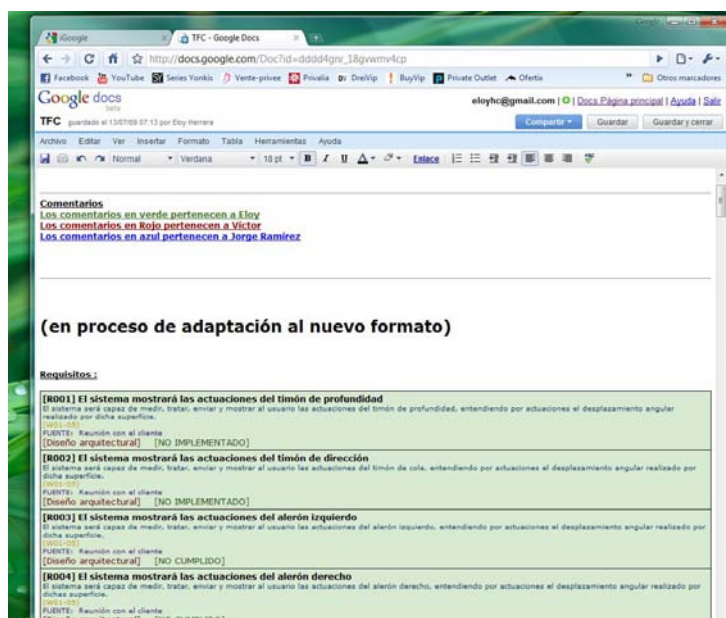
## 2.3 Herramientas utilizadas

Para asegurar la máxima uniformidad en los documentos y procedimientos llevados a cabo por los distintos departamentos es importante definir unas herramientas comunes a utilizar por éstos. Estas herramientas pueden ser, por ejemplo, de comunicación entre el personal, de elaboración de documentos, de organización temporal, etc.

En el aspecto comunicativo nuestro grupo de trabajo estableció realizar una reunión cada lunes a las 17:30. En caso de no poder asistir a estas reuniones o tener que reportar algo a otro departamento de forma urgente podremos utilizar el cliente de correo Gmail, proporcionado por Google.

Referente a organización temporal se ha creado un calendario común en Google Calendar en el que se han ido fijando los distintos hitos a alcanzar durante el desarrollo del proyecto.

Por último, al principio se estableció que la herramienta utilizada para la elaboración de la documentación sería Google Docs ya que a más de permitir la edición simultánea de un documento, nos brindaba una forma ágil de compartir todos los documentos generados por cada departamento. Aunque debido a las limitaciones de ésta herramienta decidimos migrar todos los documentos desarrollados en Google Docs a Microsoft Office 2007. En la **Figura 2.1** podemos observar el proceso de migración desde Google Docs a Word.



**Figura 2.1:** Proceso de adopción del nuevo formato.

Los compiladores del código del microcontrolador y el resto de hardware están referenciados en el documento de planificación (**ANEXO 1**).

## 2.4 Elaboración de *templates* para los documentos

Para asegurar que todos los departamentos utilizan un mismo formato para redactar su documentación es importante que ya desde la etapa de planificación se consolide un template genérico que entregar a todos los departamentos como modelo a seguir.

El template que nosotros establecimos en la etapa de planificación acabó siendo descartado en etapas posteriores debido a lo limitado de éste. Finalmente hemos acabado utilizando el template que el grupo Icarus utiliza para su documentación interna. Este nuevo template a adoptar provocó un retraso en la planificación programada.

El template antiguo puede ser consultado en el **ANEXO 1**.

## 2.5 Trazabilidad de requisitos y procesos

Como anteriormente hemos mencionado un proyecto aeronáutico se lleva a buen puerto dividiéndolo entre diferentes procesos. Estos procesos, como se muestra en la **Figura 2.2**, transcurren en paralelo y suelen durar tanto como el mismo proyecto. Por ejemplo, la verificación en contra de lo que pueda parecer, empieza junto a la planificación y no acaba hasta que el proyecto se da por terminado.

En nuestro caso solo se ha hecho la división entre diseño y seguridad por no contar con más personal. Pero si analizamos un caso real, como los que aborda el [3], nos daremos cuenta que detrás de cada uno de los documentos a redactar se encuentra todo un equipo cuya tarea dentro el proyecto consiste en una fase de uno de los procesos y la redacción de un documento justificativo del trabajo.

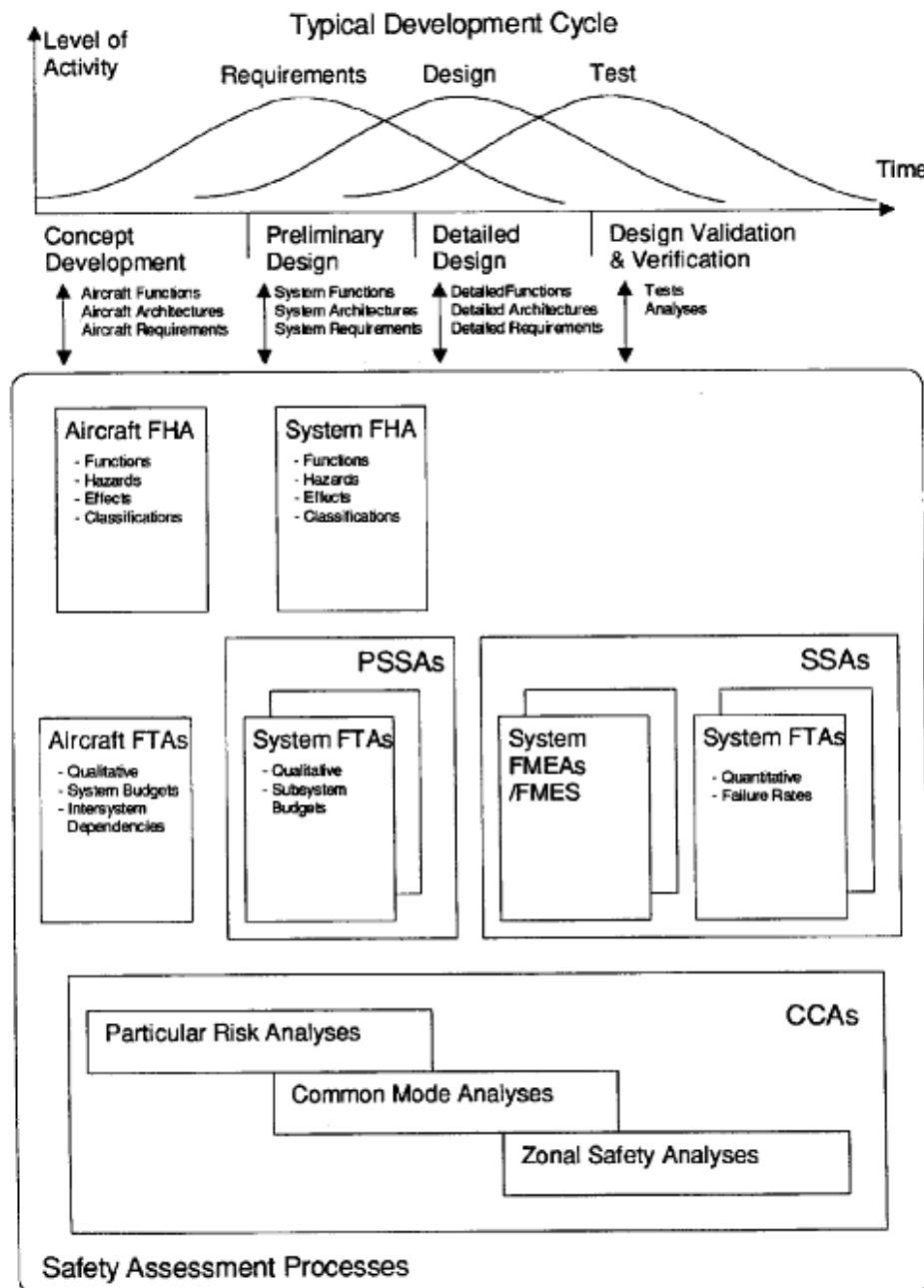
El problema está en que estas tareas no son independientes, sino que están interrelacionadas siendo “víctimas”, por así decirlo, de la necesidad de trazabilidad. Los departamentos reciben unos documentos de entrada, normalmente simplemente son una tabla de requisitos. Estas entradas son la base del trabajo a reflejar en el contenido del documento a redactar. Así mismo, en el transcurso del proceso también suelen derivarse un conjunto de requisitos o reportes de salida. Estos documentos serán a su vez reportados al resto de procesos creando así una especie de espiral de desarrollo.

Para que este sistema de organización funcione debemos definir en el documento de planificación como vamos a realizar la gestión de nuestros requisitos y documentos.

Lo más importante es asegurar que todos los procesos estén interconectados entre ellos mediante la trazabilidad de documentos. Que cuando por ejemplo, del FHA se derive un requisito, éste llegue a diseño sabiendo de dónde

procede, cuando ha sido redactado y en qué consiste. En diseño será dónde efectúen los cambios necesarios en el diseño arquitectural o detallado.

Lo mismo sucederá en verificación, dónde el departamento de diseño deberá estar continuamente informado de todos los avances que se hagan. Recibiendo tanto reportes de las pruebas pasadas como de las fallidas. Cuando reciban el informe de una prueba fallida, podrán ver los motivos del fallo y corregirlo. Una vez corregido, como si de un círculo se tratase se volverá a enviar a verificación el diseño para ver si esta vez si que satisface las condiciones de safety descritas.



**Figura 2.2:** Esquema de los procesos de un proyecto.

## 2.6 Gestión de configuración

En [3] capítulo 9, se hace referencia a la gestión de configuración. En este apartado se deben definir todos aquellos procesos que tengan a ver con la configuración del proyecto y la gestión de las diferentes tareas y resultados. Este es un apartado muy extenso que por motivos de tiempo y capital humano nosotros hemos limitado a:

- Gestión de requisitos.
- Gestión de documentos.
- Lista de requisitos.
- Gestión de tareas.

### 2.6.1 Gestión de requisitos

A diferencia de los grandes proyectos aeronáuticos dónde el registro de requisitos se realiza mediante un software especializado en la gestión de requisitos, el cual asegura una perfecta trazabilidad entre procesos, nosotros nos hemos limitado a crear una base de datos en Excel donde recogemos todos estos requisitos. El formato de los requisitos se puede ver en la **Tabla 2.3**

**Tabla 2.3:** Ejemplo de requisito

<b>[R001] Título del requisito</b>
<i>Breve explicación del requisito</i>
<i>(RW01-001)</i>
<i>FUENTE: Aquí indicaremos la fuente</i>
<i>[Madurez] [Estado]</i>

### 2.6.2 Gestión de documentos

La cantidad de documentos relativos a un proyecto aeronáutico que se generan durante el proceso de planificación, diseño o seguridad puede llegar a ser apabullante. A más, debemos recordar que estos documentos suelen moverse entre departamentos para asegurar la trazabilidad entre procesos. Esto implica que debemos realizar un arduo control de toda la documentación que generemos.

En nuestro proyecto definimos en el documento de planificación (**ANEXO 1**) cómo se iba a asegurar el control de esta documentación. Aunque finalmente y para acogernos a las directrices de la memoria hemos preferido descartar este tipo de gestión.

### 2.6.3 Lista de requisitos

En este apartado queda clara constancia de que en los procesos aeronáuticos y la documentación relativa a ellos nunca se da por cerrado un proceso. Los



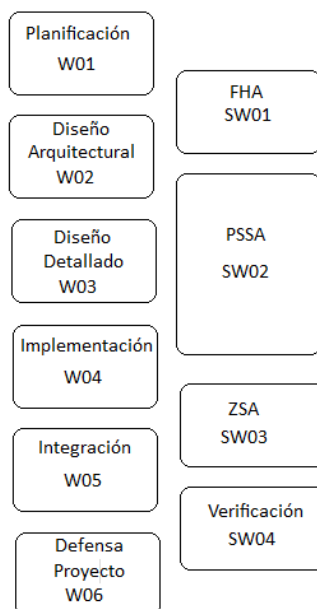
procesos están en constante evolución, en procesos más avanzados del diseño o del seguridad podemos descubrir eventualidades que nos lleven a efectuar cambios en fases más tempranas.

En el documento de planificación debemos recoger todos los requisitos generados por todos los procesos que se dan en el proyecto. Por este motivo el departamento de planificación recibe continuamente reportes con los nuevos requisitos a incluir (Esto esta automatizado en realidad). La lista con todos los requisitos, su estado y su trazabilidad puede ser consultada en el **ANEXO 11**.

## 2.6.4 Tareas y planificación

Una tarea muy importante durante la planificación de un proyecto cualquiera es hacer una lista con todos los hitos a alcanzar. Esto aun se realiza más en un proyecto aeronáutico. Sería imposible acometer un proyecto de tan bastas dimensiones sin ningún guión previo. Por esto normalmente se designa a todo un departamento a desgarnar las tareas principales lo máximo posible para no dejar nada bajo la arbitrariedad del criterio del analista o diseñador. También es necesario asignar a cada tarea un identificador para referirnos a ella en todos los documentos o en los mismos requisitos.

Nosotros tratamos de hacer algo parecido al principio del proyecto y definimos un conjunto de tareas principales que se pueden ver en la **Figura 2.3**. Notar que en la figura queda patente que las tareas de seguridad y diseño se efectúan en paralelo.



**Figura 2.3:** Lista de Tareas

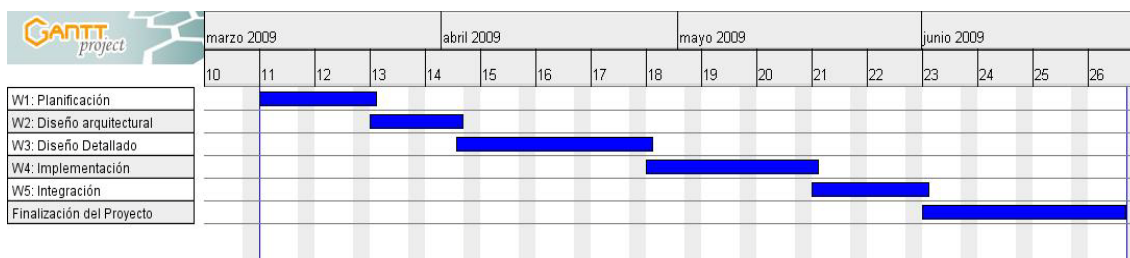
Después tratamos de desglosarlas tanto como pudimos. También como se puede ver asignamos un identificador a cada tarea, la asignación de estos

identificadores se hizo siguiendo estas pautas que están recogidas en el documento de planificación (**ANEXO 1**).

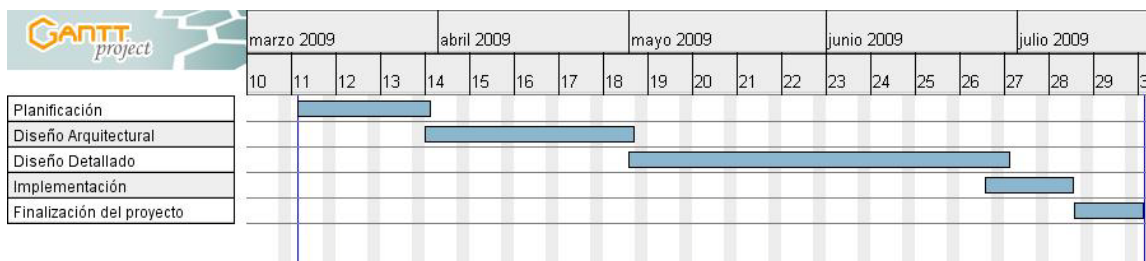
La lista completa de tareas puede consultarse en el **ANEXO 1**. Aunque debemos mencionar que estas tareas fueron las previstas antes de la elección del diseño final, por lo que incluye multitud de referencias al sensor a implantar en el motor. A más se incluyen tareas que fueron canceladas a medida que evolucionaba el proyecto por considerarlas poco útiles o por no tener la información suficiente para realizarlas.

Referente a la planificación y los plazos para los diferentes hitos tampoco han cumplido finalmente con lo previsto. La gran cantidad de retrasos sufridos junto a que la elección del diseño final se demoró mucho nos ha hecho variar en gran medida la planificación.

En la **Figura 2.4** podemos observar el diagrama de Gantt correspondiente a la planificación inicial, mientras que en **Figura 2.5** la tenemos el diagrama de gantt que resultó al final.



**Figura 2.4:** Diagrama de GANTT de la planificación inicial



**Figura 2.5:** Diagrama de GANTT de la planificación final

## 2.7 Presupuesto

En el apartado de planificación también se incluye el presupuesto. Para la realización de este se tenía una idea aproximada de los componentes que se iban a necesitar, pero no fue hasta la realización del diseño arquitectural que se definió. Al principio, se presupuestó en función de unos parámetros y utilizando una misma tienda sobre la que adquirir los productos. Finalmente, hubo ciertos cambios y se realizó un presupuesto final, que si fue cumplido. En el ANEXO 6 puede verse los dos presupuestos realizados.

## CAPÍTULO 3. FHA

### 3.1 Introducción

Según consta en el anexo A de [2], el objetivo del “Functional Hazard Assessment” (FHA) es coger las distintas funciones y identificar sus condiciones de fallo y su clasificación según impliquen una pérdida de la función o simplemente un malfuncionamiento de ésta. En el FHA también fijaremos los requisitos de safety necesarios para limitar los efectos del fallo de una función que puedan afectar a la clasificación de la condición de fallo como podemos observar en el ejemplo de la **Figura 3.1**.

Existen dos niveles dentro de un FHA, el nivel avión y el nivel sistema.

Function	Failure Condition
Control Flight Path	Inability to control Flight Path
Control Touchdown and Roll Out	Inability to control Touchdown and Roll Out
Control Thrust	Inability to control Thrust
Control Cabin Environment	Inability to control Cabin Environment
Provide Spatial Orientation	Inability to provide Spatial Orientation
Fire Protection	Loss of Fire Protection

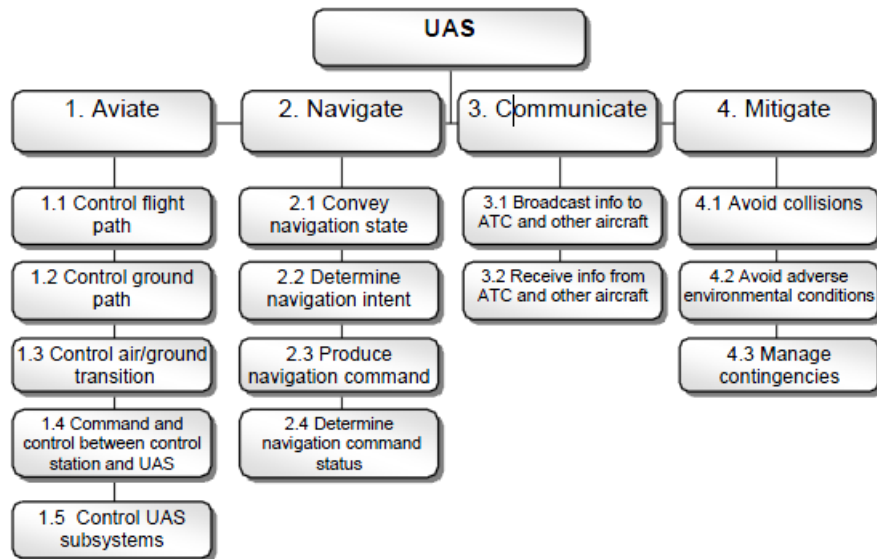
**Figura 3.1:** Ejemplo de condiciones de fallo para una función

#### 3.1.1 FHA Nivel Avión:

Para el nivel avión usaremos un estudio de safety desarrollado por la agencia espacial americana NASA. Como conclusión más importante de este informe de seguridad podemos tomar que la función que implementamos con nuestro sistema no es crítica para el avión, tal y como podemos apreciar en la **Figura 3.2**.

Esto tiene una implicación directa en el DAL a asumir, ya que ni la condición de fallo más grave que contemplamos (la pérdida del sistema) tiene efectos en el safety.

En principio, y hasta que estudios posteriores puedan decir lo contrario, al no tocar ninguna de las funciones de más alto nivel del avión podemos considerar que nuestro sistema no tiene efecto alguno en el safety del avión. Para información más detallada acerca de las funciones y su criticidad dentro del sistema aeronave consultar el documento [6].



**Figura 3.2:** Funciones de alto nivel de la aeronave

### 3.1.2 FHA Nivel sistema

En nuestro trabajo nos basaremos únicamente en éste último ya que no estamos realizando un estudio de safety de todo el avión.

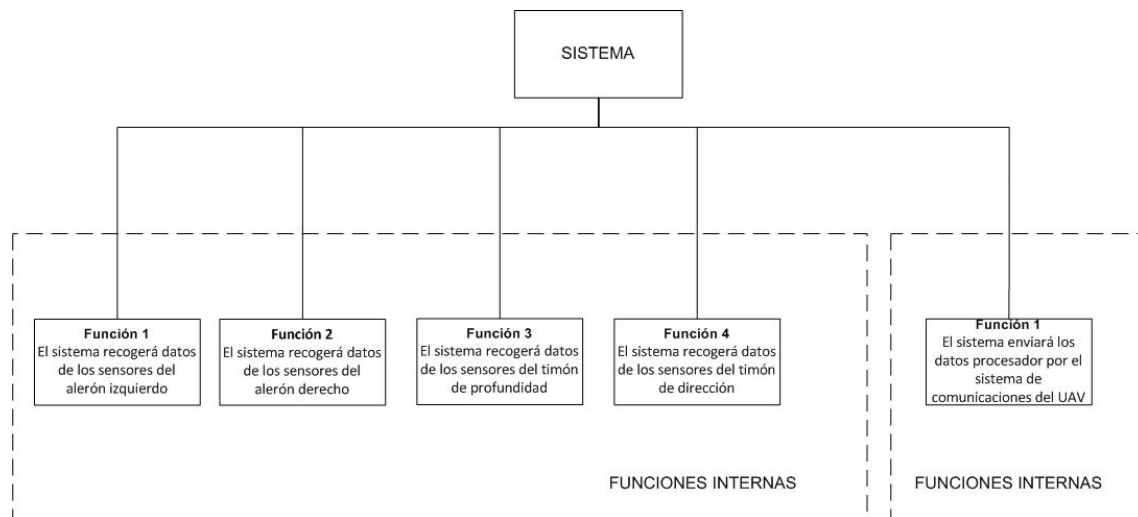
El análisis a nivel avión sobretodo se utiliza para identificar posibles combinaciones de fallos que acaben superando la clasificación de las condiciones de fallo establecidas anteriormente.

Para la realización de un FHA no deberemos utilizar documentos adicionales ya que éste constituye el primer paso en un proceso de safety.

En el proceso para identificar las condiciones de fallo y sus efectos deberemos seguir todo un conjunto de subprocesos:

#### 3.1.2.1 Identificar todas las funciones del sistema a estudiar

Una vez reunida toda la información necesaria descrita en el FHA (**ANEXO 7**) pasaremos a crear la lista de funciones. Esta lista se hace tomando como referencia la lista de funciones esperada y teniendo en cuenta la información relativa a la función recopilada en el apartado anterior. Durante este proceso encontraremos dos tipos de funciones, internas (o propias del avión) y externas (relativas a la interoperación entre aeronaves o entre éstas y tierra). Las funciones que nosotros definimos en el FHA se encuentran a modo de esquema en la **Figura 3.3**.



**Figura 3.3:** Funciones de nuestro sistema

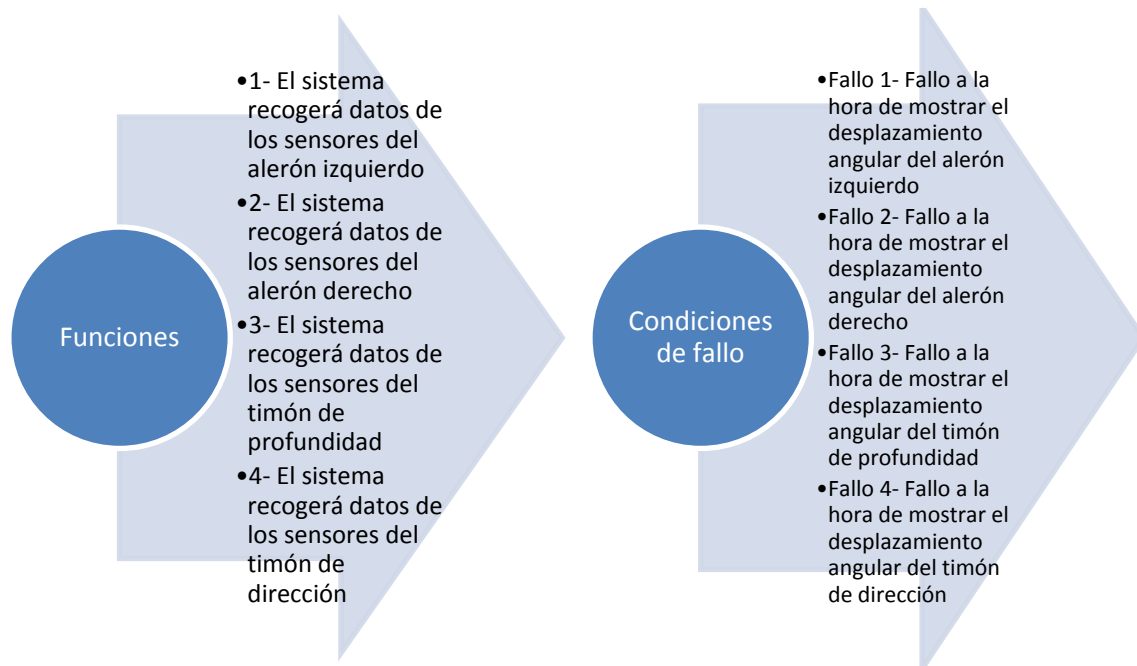
### 3.1.2.2 Identificar y describir las condiciones de fallo

Según esta recogido en el documento [2], el proceso de identificación de las condiciones de fallo empieza creando una descripción del entorno en el que trabajará la aeronave y una lista de configuraciones de emergencia. En la **Tabla 3.1** se muestra un extracto de nuestro FHA donde se describen estos parámetros.

Después el analista deberá considerar todos los sistemas que intervienen en la lista de funciones internas, la lista de funciones externas, el entorno en el que trabajará nuestro sistema y la lista de configuraciones de emergencia. Con esto el analista creará una lista de las condiciones de fallo del sistema considerando fallos tanto puntuales como múltiples en un entorno normal y en un entorno alterado. Esta lista se deberá dividir en fases de vuelo cuando la criticidad del sistema así lo aconseje. En la **Figura 3.4** se muestran las condiciones de fallo que hemos extraído del FHA (Los fallos múltiples no se contemplan en nuestro sistema al entender que cualquier fallo individual del sistema implicará una pérdida de éste).

**Tabla 3.1:** Descripción del entorno y la lista de configuraciones de emergencia.

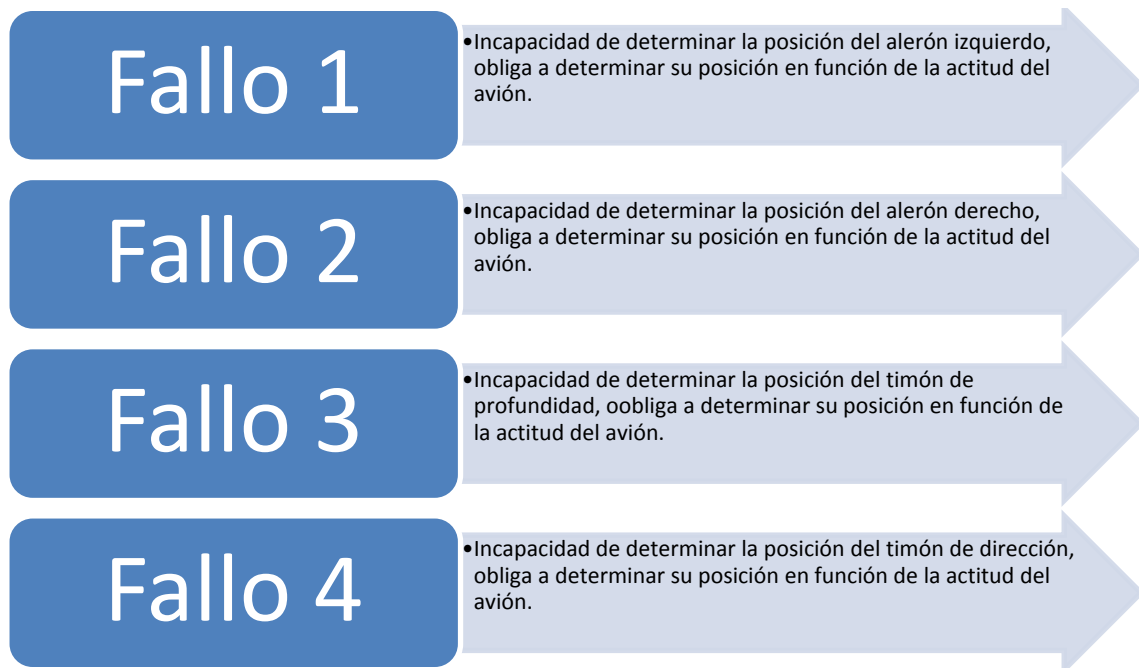
Meteorología	Condiciones anómalas
Despejado: El UAV solo se contempla para volar en condiciones de clima despejado y con poco viento. No volará en caso de que la climatología sea diferente.	<ul style="list-style-type: none"> <li>- Pérdida del sistema eléctrico</li> <li>- Pérdida del sistema de transmisión</li> <li>- Pérdida de la alimentación del sistema</li> <li>- Pérdida total del sistema eléctrico del UAV.</li> </ul>



**Figura 3.4:** Condiciones de fallo puntuales.

### 3.1.2.3 Determinar los efectos de las condiciones de fallo

Según esta recogido en el documento [2], el analista debe determinar los efectos de las condiciones de fallo en la operación normal de la aeronave o la carga. Para esto podremos consultar la opinión de expertos y gente con experiencia en la materia para ayudarnos a clasificar los efectos de las condiciones de fallo. En el caso de estar desarrollando un análisis de safety para un sistema, como es nuestro caso, el efecto a nivel de avión puede ser el mismo que a nivel de sistema o podemos considerar la combinación de efectos de otros sistemas. En la **Figura 3.5** se muestran los resultados del análisis de los efectos de las condiciones de fallo realizado en nuestro FHA.



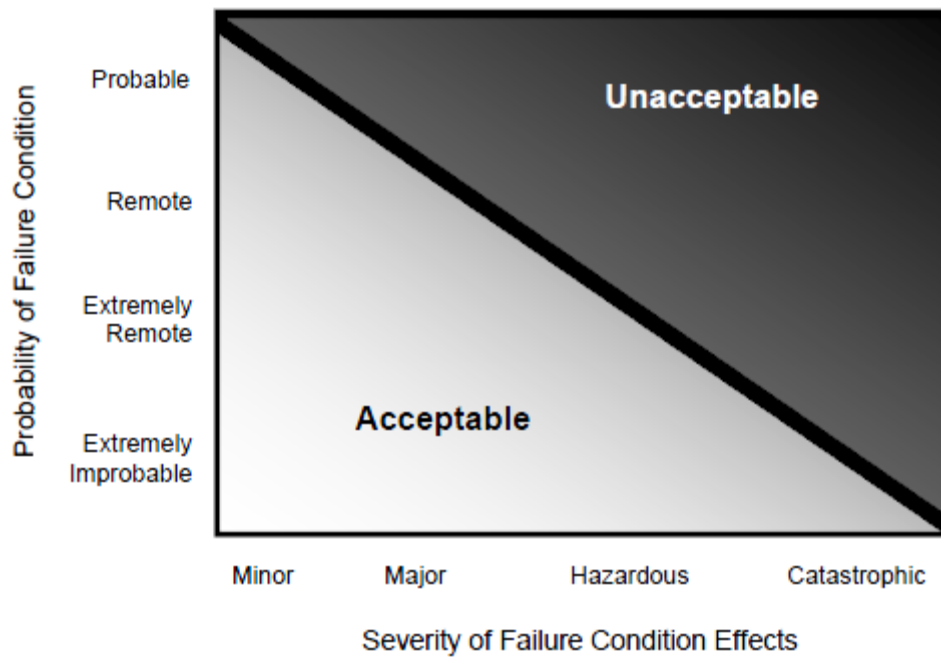
**Figura 3.5:** Efectos de las condiciones de fallo.

#### 3.1.2.4 Determinar la clasificación de los efectos de las condiciones de fallo

Deberemos movernos en el margen de lo aceptable siguiendo un criterio como el de la **Figura 3.6**, donde los fallos más críticos deberán ser extremadamente posibles mientras que los menores tendrán más margen para producirse. Deberemos clasificar el sistema dentro de cualquiera de estos niveles descritos en [2]:

- Catastrophic
- Severe-major
- Major
- Minor
- No safety effect

La determinación de esta clasificación viene dada por el análisis de los documentos o reportes de accidentes o incidente, la revisión del documento de quía regulado, usando experiencias de diseño anteriores y consultando con expertos y gente con experiencia en el campo en el que se mueve el sistema.



**Figura 3.6:** Criterios de aceptación

En principio, al ser un sistema añadido que no estaba dentro de las previsiones del UAV, se consideró que los efectos de un fallo no provocaba ningún efecto de Safety en el avión, o que era un efecto menor. Sin embargo, al ser un sistema que todos los aviones reales están obligados a llevar, consideramos que el UAV necesitaría tarde o temprano de un sistema como el que se está desarrollando. A partir de ahí, se decidió que el sistema adquiriera una clasificación del Failure Condition Effect de nivel Mayor (System Development Assurance Level - C). Debido a la arquitectura del sistema, el sistema se considera que falla cuando cualquiera de los 4 posibles fallos de sistema aparece. Eso lleva a que cada fallo debe tener la misma clasificación del Failure Condition Effect.

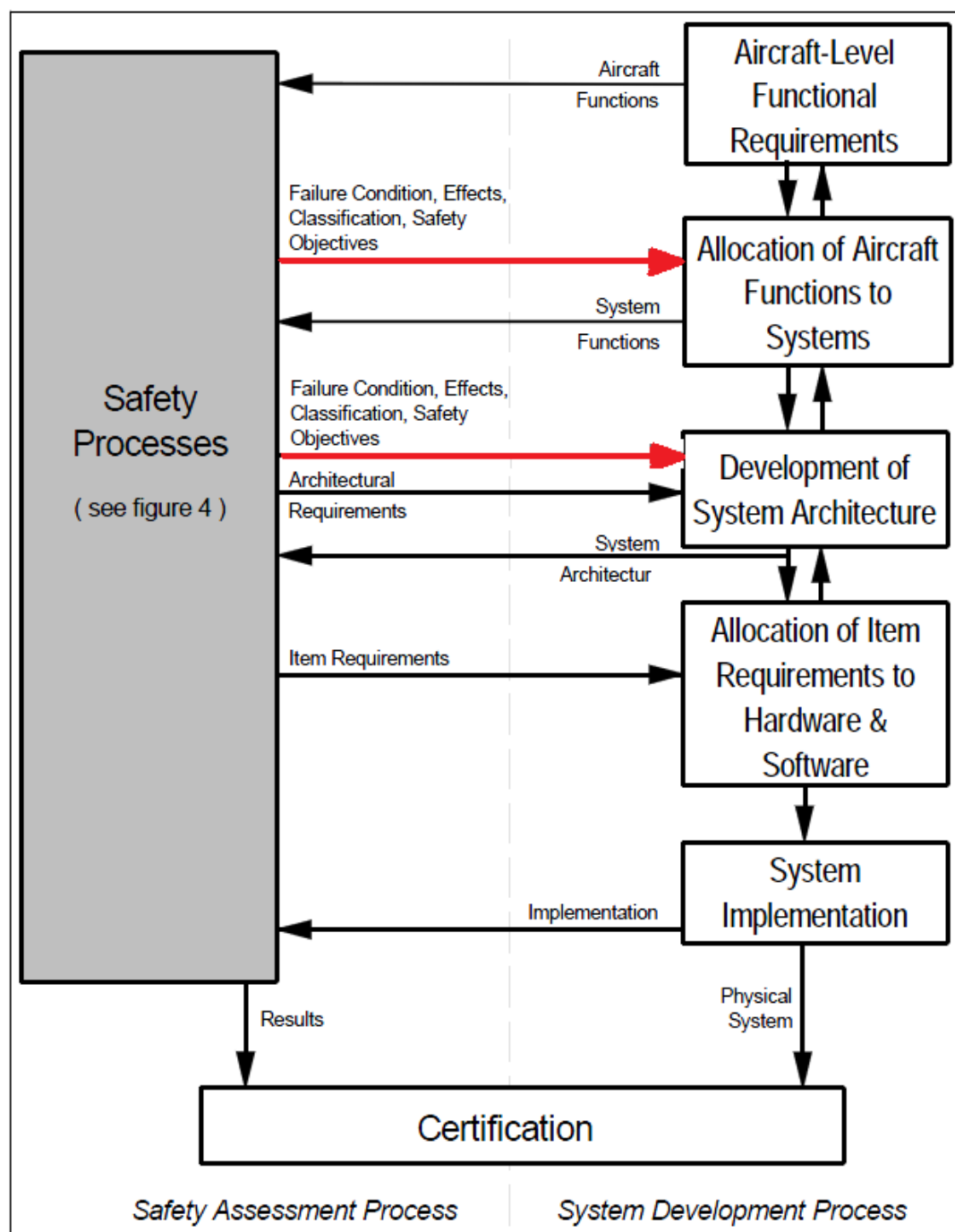
### 3.1.2.5 Asignar requisitos posibles derivados de estas condiciones de fallo

Para cada condición de fallo el analista deberá asignar posibles requisitos del sistema y requisitos de diseño. La trazabilidad de este proceso se centraliza casi su totalidad en este apartado. Después de haber hecho los estudios anteriores reveremos redactar una lista de requisitos que se reportarán al proceso de diseño. Estos requisitos serán contemplados en el diseño arquitectural y detallado.

El proceso del FHA se realiza en paralelo a los de diseño como hemos mencionado varias veces en este documento. Son ciclos iterativos que transcurren en paralelo donde es constante el envío de reportes entre los departamentos. Desde el proceso de safety enviamos requisitos a diseño y desde diseño se nos reportan los diferentes cambios realizados para satisfacer estos nuevos requisitos.



En la **Figura 3.7** podemos ver este flujo representado por una flecha roja.



**Figura 3.7:** Trazabilidad entre procesos

## CAPÍTULO 4. DISEÑO ARQUITECTURAL

### 4.1 Introducción

La arquitectura del sistema define las bases de la estructura que se va a seguir en el desarrollo del sistema. La elección de esta arquitectura está totalmente basada en el estudio de las necesidades del sistema. El Anexo 2 concreta y reúne los pasos realizados en el diseño arquitectural del sistema. Para ello, se realizó un estudio preliminar de las características generales de la plataforma sobre la que irá localizada el sistema. El conocimiento de esas características permite conocer las limitaciones y discernir entre las diferentes posibilidades, reduciendo el número de estas considerablemente. Así, tras discernir entre las distintas posibilidades, se puede llegar a unas conclusiones con las cuales será más sencillo realizar finalmente un diseño detallado.

A la hora de realizar el proceso de diseño arquitectural, se planteó la posibilidad de desarrollarlo directamente junto al proceso de diseño detallado. Sin embargo, la separación del diseño arquitectural del detallado era preferible por la realización de los documentos de seguridad. Estos documentos tienen una relación muy específica con cada parte del diseño, y por tanto era necesario realizar una separación de las diferentes partes del diseño. El trabajo en sí ya conlleva una complicación elevada al tener que realizarlo conjuntamente con los procesos de seguridad, y por tanto era mejor separar los dos bloques importantes de diseño.

Será muy importante remarcar la relación entre esta parte inicial del diseño, y el *Functional Hazard Assessment* (FHA), el documento de seguridad inicial realizado antes de iniciar el diseño.

### 4.2 Descripción del UAV

El UAV es la plataforma sobre la que se diseña el sistema y por tanto, es indispensable el estudio de este. En el ANEXO 2: Diseño Arquitectural se detalla con más precisión todos los datos del UAV. Sin embargo, aquí se realizará una pequeña descripción de la plataforma.

#### 4.2.1 Información general

El UAV es un vehículo aéreo no tripulado modelo SHADOW-MK1. El UAV, perteneciente al grupo ÍCARUS dentro de la universidad UPC, se utiliza como sistema prototipo de UAV para investigación. Todavía se encuentra en fase de desarrollo y no ha sido probado en vuelo (a fecha de julio de 2009). Las características generales del UAV pueden verse en la Tabla 1 del ANEXO 2.

La mayoría de los datos del UAV se han obtenido a partir de varias tomas de contacto con el propio vehículo. Además, han sido muy importantes las reuniones y contactos con los responsables del grupo ÍCARUS, que permitían resolver las dudas más importantes planteadas y además facilitaban ideas para poner en práctica a la hora del desarrollo. Precisamente, se obtiene del grupo el manual del SHADOW, con lo cual pudimos precisar mucho en la información del sistema (véase [1]). La **Figura 4.1** muestra una imagen del modelo SHADOW-MK1.

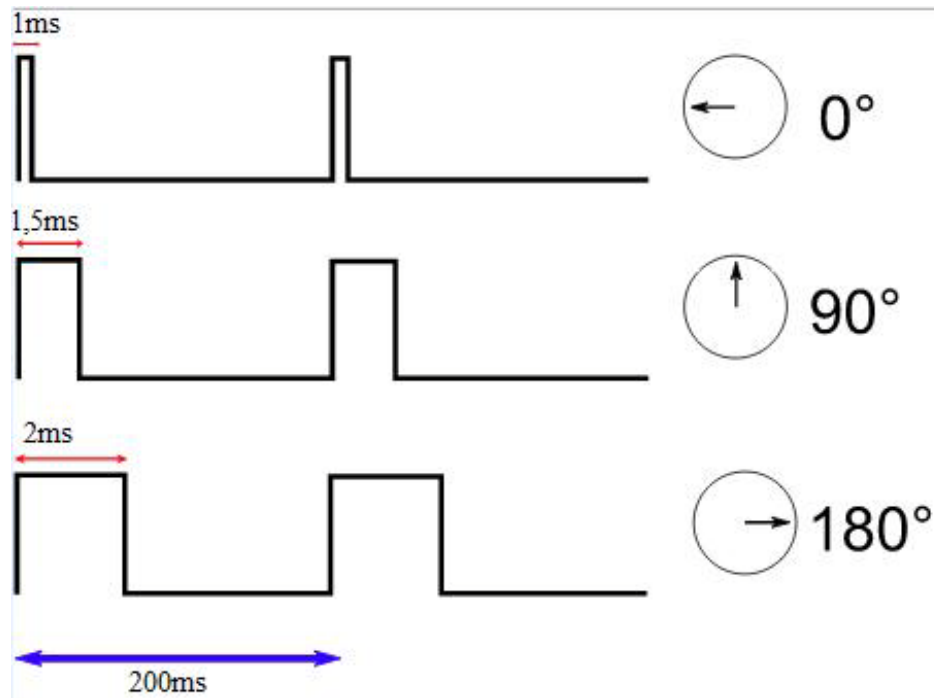


**Figura 4.1:** Fotografía del modelo SHADOW-MK1 [1]

#### 4.2.2 Superficies de control del UAV

En el estudio del UAV se comprueba que las superficies de control principales del UAV se componen de alerón izquierdo, derecho, timón de profundidad y timón de dirección. Las cuatro superficies están controladas mediante el piloto automático del UAV, y el actuador principal de estas es un servomotor (para cada una, salvo el timón de dirección que tiene dos).

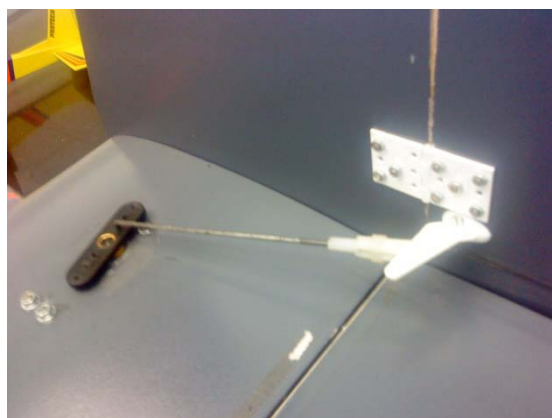
Los servomotores son motores capaces de colocarse y mantenerse en la posición que deseamos. Los servomotores (o simplemente servos a partir de ahora) estándar son principalmente analógicos, y tienen un rango de funcionamiento limitado (ligeramente superior a  $180^\circ$ ). En el ANEXO 2, se detalla con más precisión el funcionamiento de los servomotores como sistemas actuadores, sin embargo, lo más importante es entender que los servos funcionan mediante la recepción de pulsos, y que la duración de los pulsos determina la posición que debe adquirir el servo. La **Figura 4.2** es un ejemplo de la relación entre la duración del pulso y su posición en diferentes casos.



**Figura 4.2:** Variación de la posición del servo en función de la duración del pulso

El UAV emite los pulsos a través de un generador de señales que tiene una resolución de 50  $\mu$ S. Esa resolución es importante conocerla ya que tras el estudio de las diferentes partes, es la que tiene la peor resolución de todo el sistema y por tanto, la que nos servirá para escoger la resolución de nuestro sistema

El servo se conecta a la superficie de control mediante una varilla, que convierte el movimiento angular del servo en un desplazamiento lineal en la varilla. Esta a su vez se conecta a la superficie de control convirtiendo el desplazamiento lineal en un desplazamiento angular. En la **Figura 4.3** puede apreciarse como está conectado el servo y la superficie, en este caso el timón de dirección.



**Figura 4.3:** Conexión entre el servo y la superficie de control

Los detalles de la localización de cada servo y sus características se encuentran en el ANEXO 21.

### 4.2.3 Sistema de alimentación

El UAV funciona mediante una serie de baterías de 5 V a las que se puede acceder para sacar la alimentación necesaria para nuestro sistema. El ANEXO 2 ofrece más información al respecto.

### 4.2.4 Sistema de comunicaciones

El proyecto del UAV tiene fases pendientes de finalizar, y una de ellas es la implantación del sistema de buses y comunicaciones definitivo. El grupo de desarrollo MAREA dentro del ICARUS Team es el encargado de desarrollar el sistema de Comunicaciones. Actualmente, el sistema se encuentra en arquitectura PC y no está adaptado al UAV. En el futuro está previsto traspasar la arquitectura PC a una arquitectura de microcontroladores embarcados capaz de gestionar el sistema directamente desde el UAV. El funcionamiento se realizará mediante la interconexión de nodos de una red de Ethernet. Para más información, consultar [4] y el ANEXO 2.

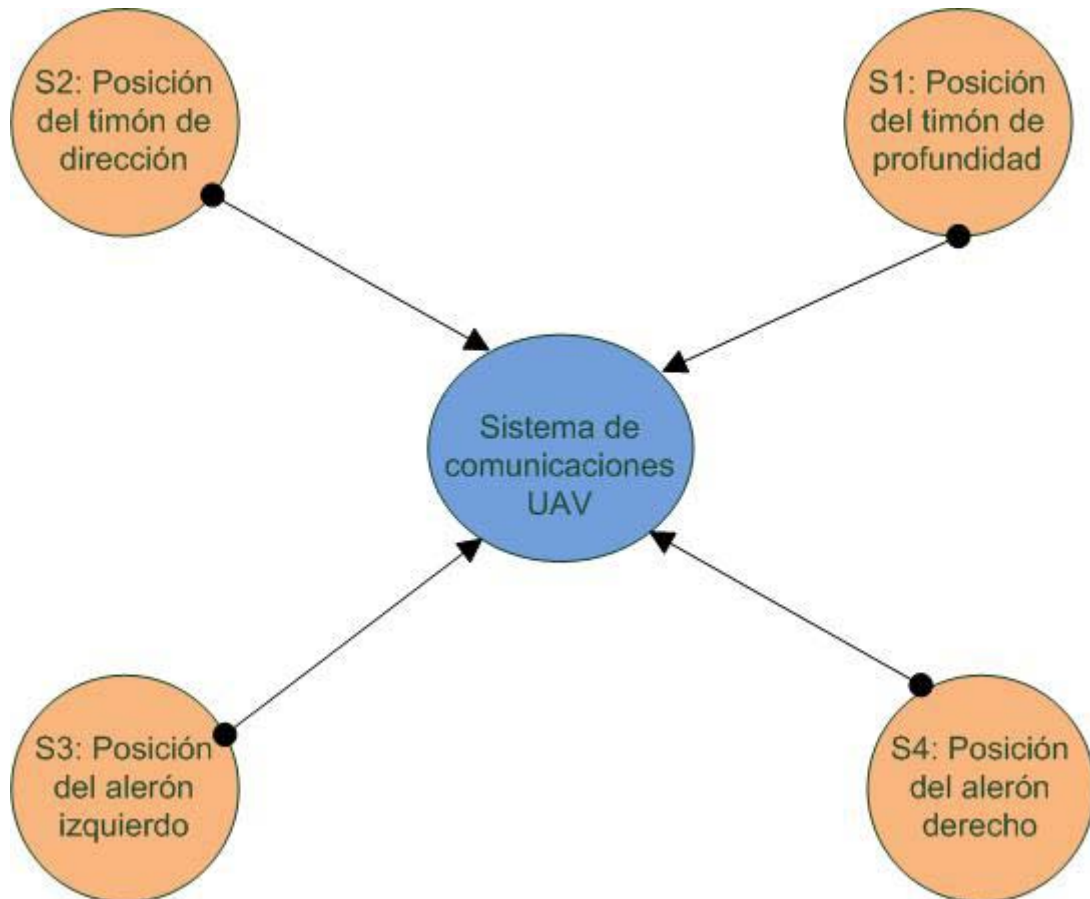
## 4.3 Diseño De La Arquitectura

Tras el estudio de las diferentes características del UAV se toman una serie de decisiones, apoyándonos siempre en las consideraciones de seguridad que llegan desde el FHA.

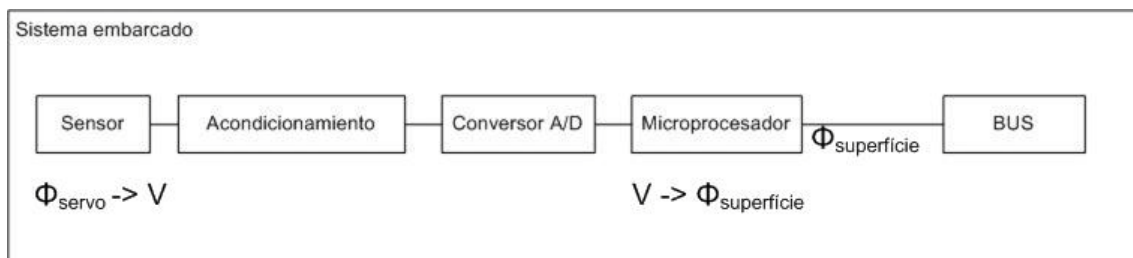
### 4.3.1 Decisiones iniciales

La primera decisión importante realizada se basa en la división del sistema en cuatro subsistemas totalmente independiente entre ellos. Cada subsistema será el encargado de monitorizar los datos de la superficie a la que están designados. Se asigna así un subsistema a cada superficie a controlar. Cada uno emitirá su posición utilizando el sistema de comunicaciones del UAV y a partir de ahí se recogería la información que emiten estos sensores para mostrar la posición de la superficie. La **Figura 4.4** muestra un esquema de cómo sería el sistema.

Sobre la arquitectura de cada subsistema, se decide utilizar una arquitectura estándar para cualquier tipo de sistema de sensores, formado principalmente por un sensor y su acondicionamiento, un conversor analógico-digital, y un microcontrolador que trate los datos para luego enviarlos por un sistema de transmisión. La **Figura 4.5** muestra un diagrama de bloques de cómo será cada subsistema.



**Figura 4.4:** Esquema del sistema



**Figura 4.5:** Diagrama de bloques de los subsistemas

### 4.3.2 Resolución del sistema

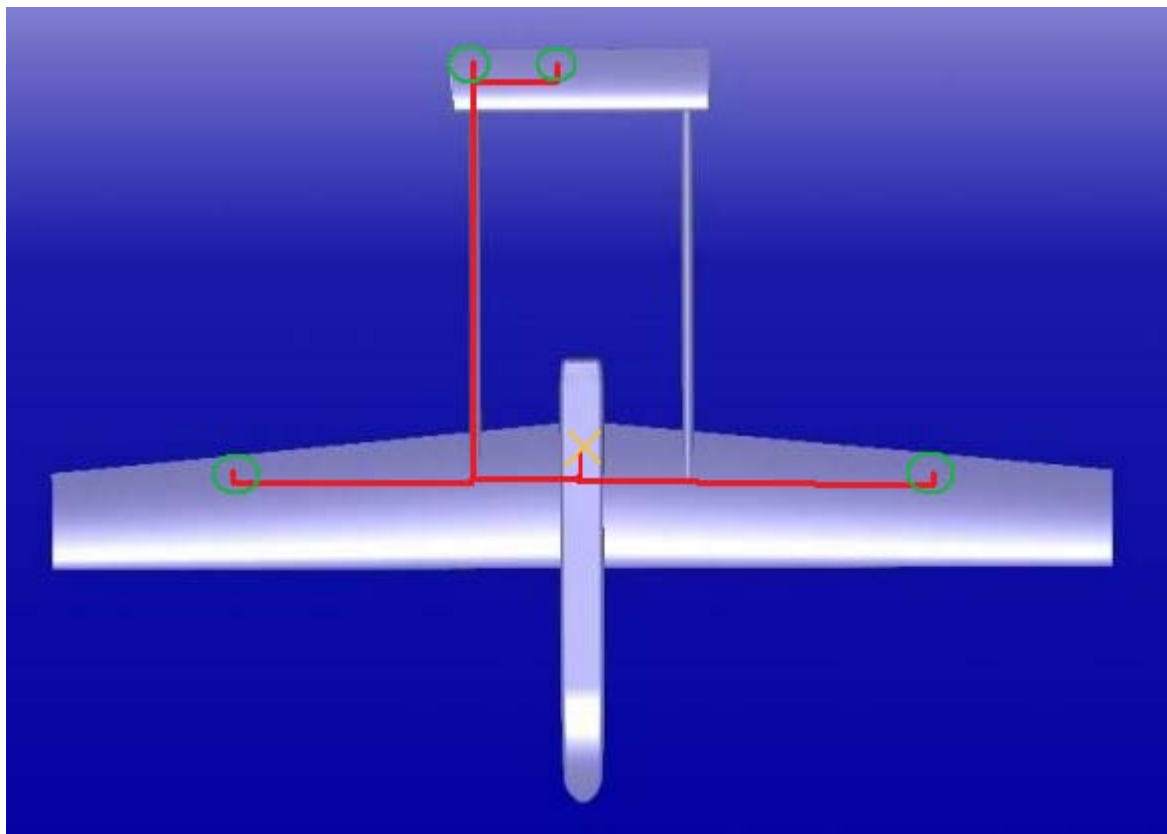
La resolución del sistema estará escogida en función de la peor resolución que podemos obtener del sistema. Tal y como está explicado en la descripción del UAV en este mismo capítulo, la peor resolución la ofrece el generador de los pulsos del servo. Así, podemos conocer la resolución que tendrá cada subsistema, pues el rango dinámico está calculado a partir de esa resolución (ver ANEXO I). La **Tabla 4.1** indica las resoluciones para cada subsistema (la obtención de esta viene detallada también en el ANEXO 2).

**Tabla 4.1:** Resolución de los diferentes subsistemas

Superficie (subsistema)	Resolución
Timón de profundidad (S1)	1.46°
Timón de dirección (S2)	1.46°
Alerón Izquierdo (S3)	2.375°
Alerón Derecho (S4)	2.333°

### 4.3.3 Sistema de Alimentación

El sistema de alimentación procederá de las baterías de instrumentación del propio UAV, por tanto no será necesario diseñar un sistema con alimentación propia. Debido a las características del sistema, el consumo será muy bajo y por tanto el gasto energético no será un factor muy importante. Sí es importante como se llevará la alimentación a cada subsistema, y por ello se utilizará un esquema de alimentación que vaya a cada zona donde se requiera la alimentación, tal como muestra la **Figura 4.6**.

**Figura 4.6:** Distribución de la alimentación

## 4.4 Consideraciones de seguridad durante el diseño arquitectural

El diseño arquitectural se ve condicionado desde el principio por los informes de seguridad. En este caso, el mayor documento de seguridad que afecta al diseño es el *Functional Hazard Assessment* (a partir de ahora FHA). El FHA del sistema considera que siempre que con que falle cualquiera de las partes principales del sistema, el sistema entero se considerará que ha fallado.

Una de las decisiones por las que realizar un sistema con subsistemas independientes es precisamente seguir el esquema utilizado en el desarrollo del FHA, donde cada parte del sistema que monitoriza una de las superficies queda totalmente separada de la otra. Eso produce que para una mejor gestión de los procesos una relación más sencilla entre los informes de seguridad y los de diseño, sea lo más conveniente. Es un caso de optimización de los procesos de diseño y safety, para facilitar la comunicación y la interconexión entre ellos.

Nuestro diseño, debido a la simplicidad del sistema y que es un prototipo, no cuenta con redundancias para controlar que si falla uno de los subsistemas, este siga funcionando. Esto se basa en el FHA del avión realizado por la NASA e incluido en el **ANEXO 7**. La función de monitorización realizada no aparece en el FHA y por tanto, se considera que no tiene una criticidad catastrófica y por consiguiente no era necesario hacer un sistema redundante, con su consecuente aumento de presupuesto, no dispuesto a ser asumido. Se podría considerar realizar una redundancia de los sistemas a pesar del FHA y el presupuesto, sin embargo, debido a parámetros como el espacio donde actuará, la complejidad que otorga, y la consideración de que el UAV es un prototipo, se decidió que no era necesario.



## CAPÍTULO 5. PSSA

### 5.1 Introducción

Según esta recogido en el documento [2], el “Preliminary System Safety Assessment” (PSSA) consiste en examinar la arquitectura del sistema propuesto determinando como los fallos en ésta nos pueden llevar a los peligros identificados en el “Functional Hazard Assessment” (FHA) y cómo los requisitos del FHA pueden ser identificados. El proceso del PSSA interactúa y está asociado con el de diseño, yendo los dos unidos a través del proceso iterativo que es una espiral de diseño.

Para cada sistema analizado el PSSA reportará un conjunto de condiciones de fallo que pueden llevarnos a los peligros descritos en el FHA. Para la elaboración de un PSSA deberemos tener en cuenta el diseño de la arquitectura del sistema, su complejidad, la peligrosidad de sus condiciones de fallo y el tipo de función que nuestro sistema realiza dentro de la aeronave. Para esto, un requisito primordial será el FHA.

Para determinar cómo los fallos en el sistema pueden llevar a los peligros destacados en el FHA y como los requisitos del FHA pueden ser satisfechos, deberemos completar una lista con los requisitos de safety del sistema. Para la elaboración de ésta lista necesitaremos como documentos de entrada:

- FHA (**ANEXO 7**)
- Un Common Cause Analysis (CCA) provisional.  
*Este documento no ha sido realizado ya que todos los análisis a hacer son de nivel aeronave y como anteriormente hemos indicado en el Capítulo 3, para este nivel tomamos como válido el estudio [6].*
- Una descripción de la arquitectura del sistema (y su justificación). (**ANEXO 2**)
- Una lista del equipo a utilizar y sus respectivas funciones.
- Las relaciones entre los distintos sistemas.

Después deberemos determinar si nuestra arquitectura y/o diseño permite cumplir con estos requisitos. Para esto deberemos tener en cuenta:

- Un FTA dónde se muestre como los diferentes fallos por si mismos o mediante una combinación de ellos pueden llegar a cumplir con la condición de fallo.
- Una identificación de posibles independencias entre los fallos.
- Un FTA dónde se muestre que requisitos y objetivos relacionados con las condiciones de fallo se pueden verificar y cuáles no.

- La determinación de los procesos y tiempos de mantenimiento de los diferentes subsistemas.
- La determinación de un DAL para los diferentes subsistemas.

Y por último trazar estos requisitos al diseño de los pequeños subsistemas que forman nuestro sistema (software y hardware). Para esto deberemos realizar:

- Una lista de las condiciones de fallo actualizadas.
- Los requisitos de safety para cada subsistema.
- Los DALs del software y el hardware.
- Determinar las tareas de mantenimiento y sus tiempos.

## 5.2 Lista de requisitos de safety del avión

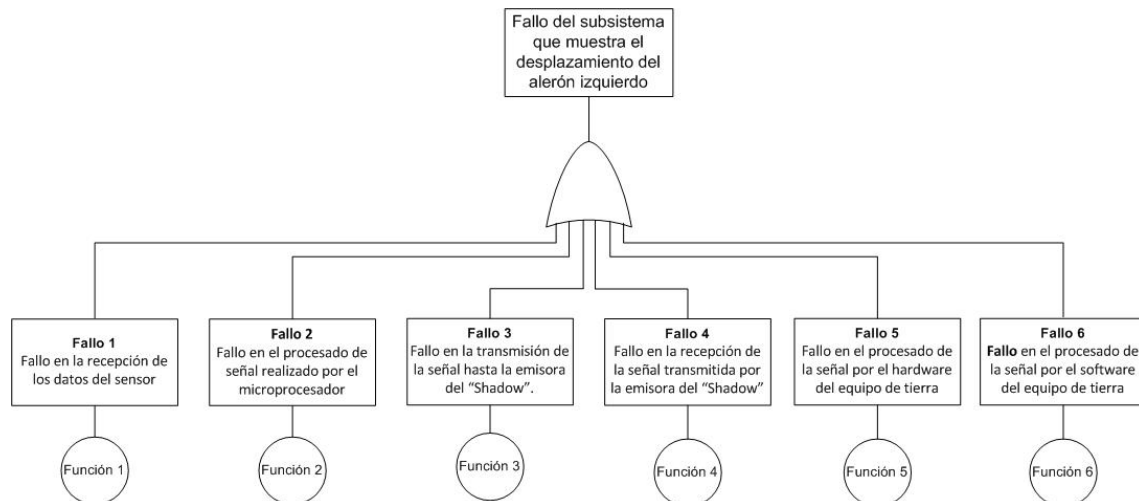
Según esta recogido en el documento [2], en la redacción del FHA creemos una lista de funciones a las que atribuimos unas condiciones de fallo y unos requisitos derivados de safety para estas condiciones de fallo. Ahora en el PSSA deberemos retomar estas condiciones de fallo primarias del FHA y desgranarlas en condiciones de fallo de los diferentes subsistemas.

Adjuntaremos también un *Fault tree analysis* (Consultar [2] apartado 4.1 para una explicación detallada de cómo llevarlo a cabo) donde mostraremos como las distintas condiciones de fallo desgranadas en el PSSA pueden llegar a producir la condición de fallo mayor proveniente del FHA.

Deberemos también completar unos requisitos de safety para estos subsistemas que nos aseguren el no incurrir en ninguna de las condiciones de fallo destapadas en el PSSA. Reportaremos al departamento de diseño estos requisitos para que los tengan en cuenta en el proceso de diseño del sistema. Podemos ver un extracto de estos requisitos en la **Tabla 5.1**. En la **Figura 5.1** podemos ver el *Fault Tree Analysis* para el alerón izquierdo a modo de ejemplo, la resta de esquemas se podrán consultar en el PSSA adjunto a este documento.

**Tabla 5.1:** Extracto de tabla de requisitos derivados del PSSA.

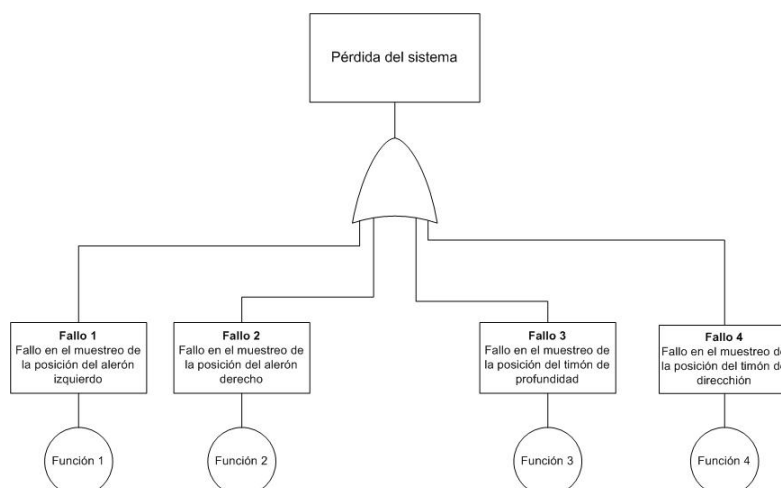
<p><b>[RS006] La probabilidad de fallo en la recepción de datos del sensor debe estar clasificada como “Major Failure Condition”</b></p> <p>SW02-02</p> <p>FUENTE: Realización del PSSA</p> <p>[Diseño arquitectural] [NO IMPLEMENTADO]</p>
<p><b>[RS007] La probabilidad de fallo en el procesamiento de señal realizado por el microprocesador debe estar clasificada como “Major Failure Condition”</b></p> <p>SW02-02</p> <p>FUENTE: Realización del PSSA.</p> <p>[Diseño arquitectural] [NO IMPLEMENTADO]</p>
<p><b>[RS008] La probabilidad de fallo en la transmisión de la señal hasta la emisora del “Shadow” (Cableado) debe estar clasificada como “Major Failure Condition”</b></p> <p>SW02-02</p> <p>FUENTE: Realización del PSSA.</p> <p>[Diseño arquitectural] [NO IMPLEMENTADO]</p>



**Figura 5.1:** Condiciones de fallo para el alerón izquierdo

### 5.3 Evaluar las decisiones de diseño con respecto a los requisitos y objetivos del sistema

Según esta recogido en el documento [2], en este apartado deberemos mostrar, utilizando un FTA o algún método similar, que los requisitos y objetivos asociados a las condiciones de fallo pueden ser demostrados con la arquitectura propuesta. Nosotros nos hemos valido del diagrama de la **Figura 5.2**, extraído del FHA para demostrar que cualquier fallo en uno de los subsistemas que conforman nuestro sistema conduce ya directamente a la pérdida del sistema.



**Figura 5.2:** Diagrama de pérdida del sistema

Como se observa en la **Figura 4.5** del capítulo 4, Diseño arquitectural, nuestra arquitectura se divide en cuatro subsistemas completamente independientes

entre ellos enlazados a un subsistema global. Con esto buscamos asegurar que las probabilidades de fallo no se relacionan entre sí.

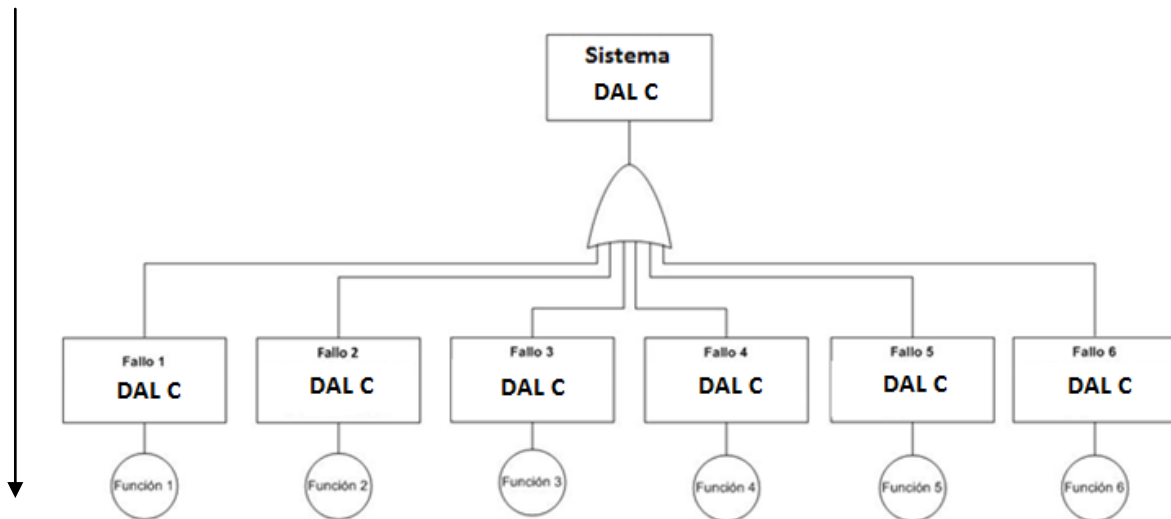
Según el documento [2] en este apartado también debemos mencionar las tareas de mantenimiento asociadas a nuestro sistema que puedan ayudar en medida de lo posible a satisfacer los requisitos de Safety. Según este mismo documento también será necesario definir un Development Assurance Level (DAL) para todos nuestros sistemas.

Para definir un DAL nos valdremos de los requisitos derivados del FHA y de los primeros compases del PSSA. Según estos requisitos nuestro sistema debe satisfacer una "*Major Failure Condition*" que como podemos ver en la **Tabla 5.2** extraída del documento [3] capítulo 3 le corresponde un DAL C.

**Tabla 5.2:** Clasificación de las condiciones de fallo según su DAL

Failure Condition Classification	System Development Assurance Level
Catastrophic	A
Hazardous/Severe Major	B
Major	C
Minor	D
No Safety Effect	E

Por este motivo, podemos ver como en la **Figura 5.3** a todos nuestros sistemas les hemos asignado un DAL C:



**Figura 5.3:** DAL de los diferentes sistemas.

## 5.4 Derivar los requisitos de “safety” al diseño de los subsistemas

Según esta recogido en el documento [2], en este punto deberemos enlazar nuestros requisitos a directrices que el departamento de diseño tendrá que tener en cuenta al elegir el hardware y software que formará parte de nuestro equipo.

### 5.4.1 Lista de condiciones de fallo actualizada que incluya unas directrices de cómo los requisitos pueden ser satisfechos.

Nuevamente, según indican las directrices de [2], para este punto elaboraremos una tabla donde figuren por este orden:

- Condición de fallo
- Requisito o requisitos derivados de la condición de fallo.
- Directrices de cómo estos requisitos pueden darse por satisfechos o no.

Incluimos en este documento, a modo de ejemplo un extracto de la tabla desarrollada en el PSSA para la elaboración de este procedimiento (**Tabla 5.3**). Para consultarla entera remitirse al PSSA adjunto.

**Tabla 5.3:** Condiciones de fallo actualizadas

Fallo en la recepción de datos del sensor	[RS006] [RS012] [RS018] [RS024]	-Comprobar que la probabilidad de fallo del sistema es inferior a una cada 10.000 horas de vuelo. Para esto nos ayudaremos de un banco de pruebas y un sistema de ciclos acelerados. -Consideraremos posible fallo del potenciómetro cuando: -Información recibida incorrecta (25%). -No recibir información. -Recibir código de seguridad.
Fallo en el procesado de señal realizado por el microprocesador.	[RS007] [RS013] [RS019] [RS025]	-Comprobar que la probabilidad de fallo del sistema es inferior a una cada 10.000 horas de vuelo. Para esto nos ayudaremos de un banco de pruebas y un sistema de ciclos acelerados. -Consideraremos posible fallo del microprocesador cuando: -Información recibida incorrecta(25%). -No recibir información. -Recibir código de seguridad (50%).
...	...	...

### 5.4.2 Lista con los requisitos de “safety” interrelacionados con el hardware y software.

Según esta recogido en el documento [2], En este punto valoraremos que partes del equipo intervienen o no en el cumplimiento de un determinado requisito. Mostraremos esta interrelación con una tabla que constará de la siguiente estructura:

- Condición de fallo.
- Requisito o requisitos derivados de esa condición de fallo.
- Software/hardware interrelacionado.

También incluimos en este documento, a modo de ejemplo un extracto de la tabla desarrollada en el PSSA para la elaboración de este procedimiento (**Tabla 5.4**). Para consultarla entera remitirse al PSSA adjunto.

**Tabla 5.4:** Requisitos de “safety” interrelacionados con el hardware y software

Fallo en la recepción de datos del sensor..	[RS006] [RS012] [RS018] [RS024]	Potenciómetro
Fallo en el procesado de señal realizado por el microprocesador.	[RS007] [RS013] [RS019] [RS025]	Microprocesador
...	...	...

### 5.4.3 Requisitos para la instalación del sistema

En [2] se nos insta a, en este punto, estudiar todos aquellos parámetros o implicaciones de la instalación de nuestro sistema en la aeronave. Se trata de elaborar un listado de requisitos que remitiremos al equipo de diseño. En nuestro caso, los requisitos que hemos creído oportuno reportar al departamento de diseño están recogidos en la **Tabla 5.5**.

**Tabla 5.5:** Requisitos para la instalación del sistema.

<b>[RS035] La señal no se verá afectada por el ruido generado en el avión.</b> SW02-02 FUENTE: Realización del PSSA [Diseño arquitectural] [NO IMPLEMENTADO]
<b>[RS036] El sistema no afectará al correcto comportamiento mecánico de las superficies de control</b> SW02-02 FUENTE: Realización del PSSA. [Diseño arquitectural] [NO IMPLEMENTADO]

[RS037] El sistema no interferirá en la aerodinámica de la aeronave

SW02-02

FUENTE: Realización del PSSA.

[Diseño arquitectural] [NO IMPLEMENTADO]

#### 5.4.4 DAL del software y el hardware

Aunque en el apartado 5.3 de este documento ya hemos hablado acerca del DAL asociado a las condiciones de fallo del sistema. Ahora el [2] indica que debemos definir un DAL para cada ítem de nuestro sistema. Para esto nos valdremos de los requisitos que hemos definido en el primer apartado del PSSA, los cuales indican que los diferentes ítems que componen nuestro sistema deben satisfacer una “*Major failure condition*”, y la **Tabla 5.2: Clasificación de las condiciones de fallo según su DAL**, donde se asigna un DAL a cada condición de fallo.

Debido a que en nuestro sistema ningún ítem es redundante y cualquier fallo en uno de ellos ocasiona la pérdida del sistema. El DAL de los ítems se corresponderá con el DAL del sistema. Que como podemos ver en la **Tabla 5.6** será un DAL C.

**Tabla 5.6:** DAL de Software y hardware

Hardware	DAL
Potenciómetro	C
microprocesador	C
Cableado	C
Emisor/Receptor	C
Hardware equipo tierra	C
Software equipo tierra	C

#### 5.4.5 Tareas de mantenimiento

Tanto el hardware como el software que forman parte del sistema carecen de defectos de degradación temporal, por esto es innecesario imputar unas obligaciones de mantenimiento para cumplir con los requisitos de safety. Además no contamos con MTBF de ningún componente del hardware como para aconsejar ninguna de estas tareas.

### 5.5 Trazabilidad de los requisitos derivados del PSSA

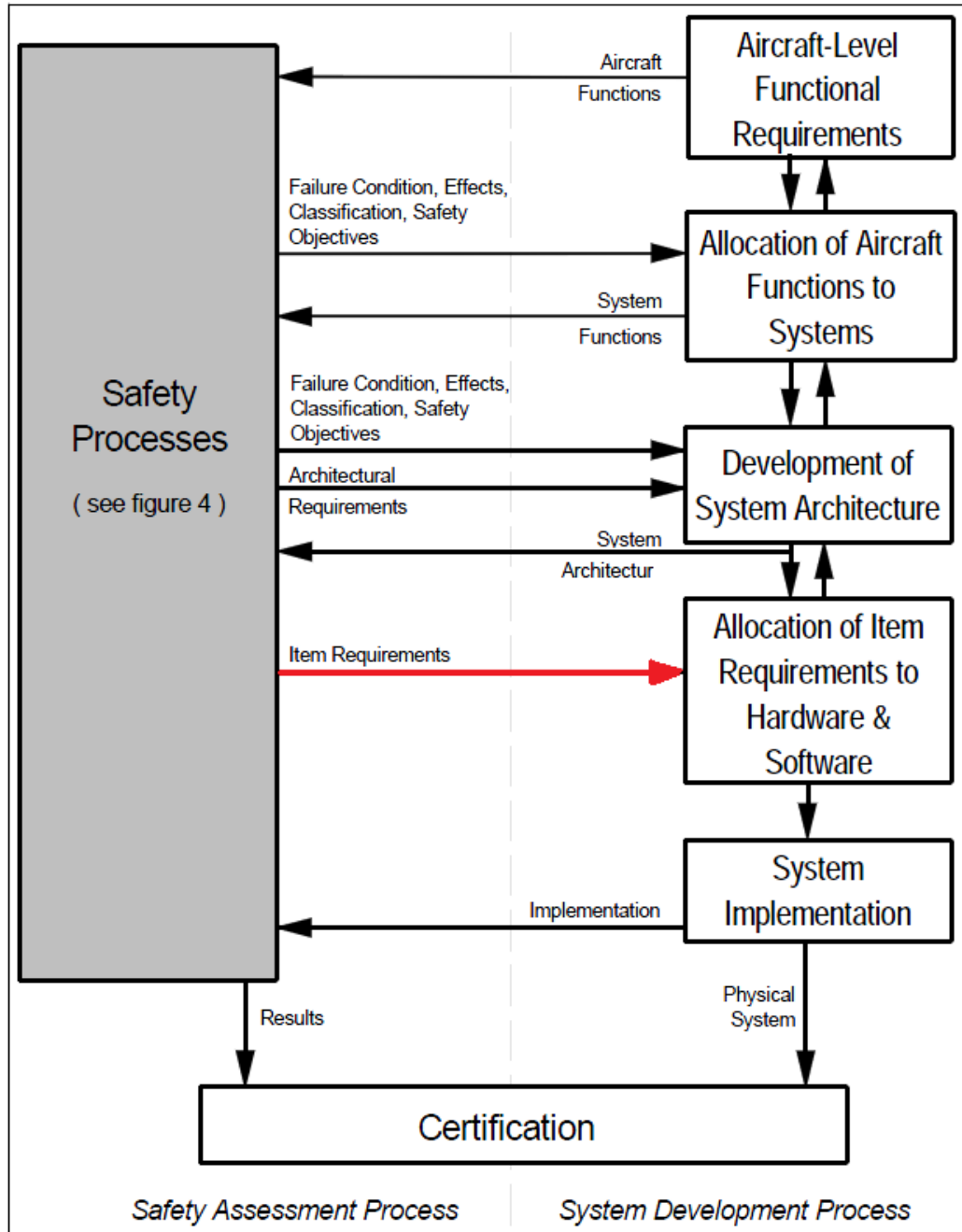
Los requisitos del PSSA brotan de dos fuentes diferentes, los requisitos derivados de desgranar las condiciones de fallo provenientes del FHA y los derivados del embarcado del sistema en el avión.

En los primeros básicamente se le asigna una condición de fallo a cada subsistema. El conjunto de estos requisitos se reportarán al proceso de diseño

para que ajusten los ítems a utilizar a los requisitos de la condición de fallo que desde safety les hemos impuesto

Mientras que en los segundos se hace un estudio de todas las implicaciones de la instalación del sistema dentro de un UAV. Estos requisitos serán tomados en cuenta por el departamento de diseño que se encargue de especificar como implantar el sistema en el avión.

En la **Figura 5.4** podemos observar el flujo aquí descrito.



**Figura 5.4:** Flujos de safety



## CAPÍTULO 6. DISEÑOS DESCARTADOS

### 6.1 Introducción

La elaboración de un proyecto de sensores conlleva un estudio de las diferentes posibilidades a tener en cuenta. La funcionalidad de los sensores es muy dinámica y hay muchas formas diferentes de medir los datos deseados de un sistema. Este proyecto conlleva la búsqueda de un sistema sensor que se adapte a los requisitos del sistema, y también a los requisitos derivados de los diferentes estudios de seguridad. Precisamente, uno de los motivos principales por los que se han tenido que descartar algunos diseños han sido los requisitos provenientes del Preliminary System Safety Assessment.

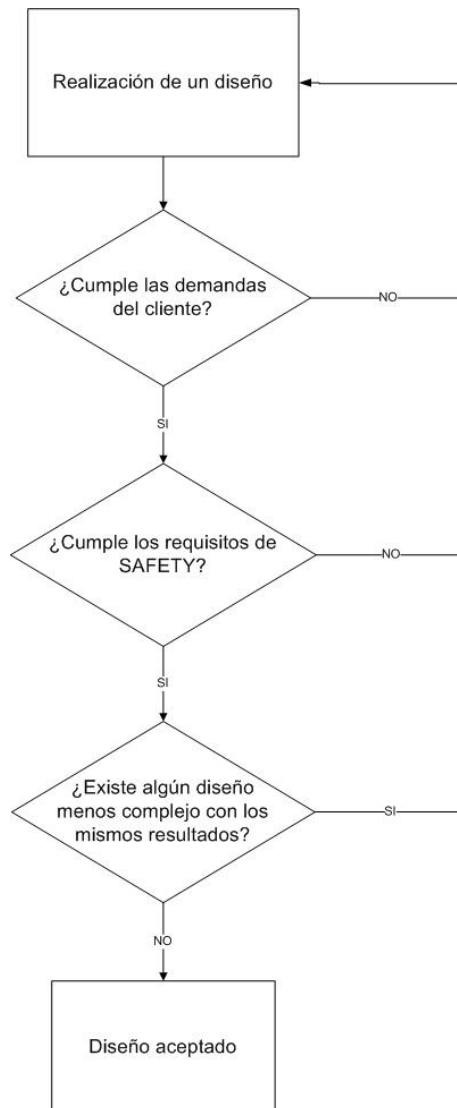
El proceso seguido para la realización de los diferentes diseños se ha basado en un proceso iterativo hasta que se ha encontrado el diseño deseado. La **Figura 6.1** muestra el algoritmo de iteraciones realizado para encontrar el diseño deseado. Este algoritmo se basa en que el diseño debe cumplir una serie de requisitos e imposiciones para poder ser aceptado.

Primero, se comprueba que el diseño realizado cumple con las demandas y requisitos del cliente. Lógicamente, si no los cumple, no hay que continuar con el proceso y se tiene que reiniciar, pues de nada sirve que cumpliera otros aspectos si no cumple la funcionalidad principal que se está buscando.

Tras esto, se hace una revisión de los requisitos derivados de seguridad. Estos requisitos, desarrollados por el departamento de seguridad, son claves pues aseguran que el sistema puede funcionar en el avión y no afectar de forma negativa sobre este. Es una forma de asegurar que nuestro sistema no puede realizar ninguna acción que comprometa la seguridad del avión.

Finalmente, es importante buscar el diseño más sencillo de todos que cumpla con todo lo anterior. Cuanto más sencillo sea un sistema, más fácil de realizar será y será más complicado que falle. La complejidad trae consigo mayores problemas y que pueden hacerse muy complicados de resolver.

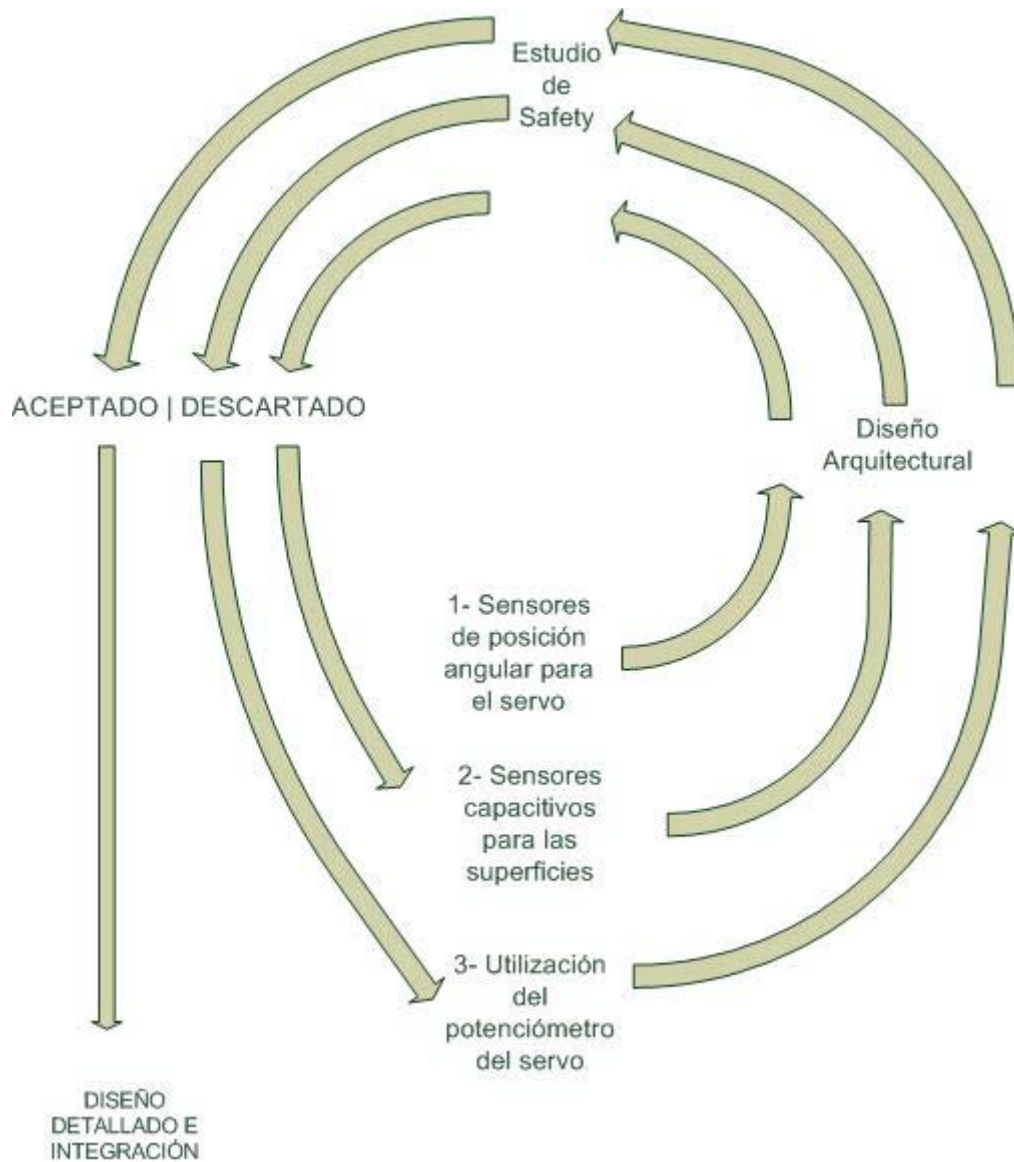
La realización de este proceso ha tenido que hacerse diversas veces hasta encontrar el diseño definitivo, pues hasta que no se han cumplido todas las condiciones, no se ha llevado a cabo el diseño detallado final.



**Figura 6.1:** Proceso Iterativo para la realización del diseño

La realización de este proyecto se vio muy afectada debido a que se tuvieron que realizar diversas iteraciones hasta llegar al diseño definitivo. Cada paso conllevaba un gasto de tiempo importante, ya que se realizaba un estudio de las posibles opciones y como llevarlo a cabo. Y no solo el realizar los diferentes estudios, sino idear las distintas posibilidades. Finalmente, tras estudiar detenidamente el sistema, se consiguió realizar un diseño que cumplía todos los requisitos de diseño y de seguridad. La **Figura 6.2** muestra el proceso iterativo realizado con los distintos diseños, y como finalmente se encontró el diseño deseado. Los diseños realizados que aparecen en la figura son:

- 1- Sensores de desplazamiento angular: Potenciómetro externo
- 2- Sensores capacitivos
- 3- Sensor interno del servo



**Figura 6.2:** Proceso de diseño iterativo hasta el diseño final

## 6.2 Sensores de Desplazamiento Angular: Potenciómetro

### 6.2.1 Tecnología del sensor

Se planteó la utilización de un sensor de desplazamiento angular externo para medir la posición del servo, y extrapolarla a la superficie. Existen diferentes tipos de sensores angulares: Potenciómetros, magneto-resistivos, encoders, LVDT's y de efecto Hall, entre otros. Por su sencillez, por la resolución que debía tener el sistema, y por su precio, de entre todas las opciones se decide coger un sensor de resistencia variable, es decir, un potenciómetro.

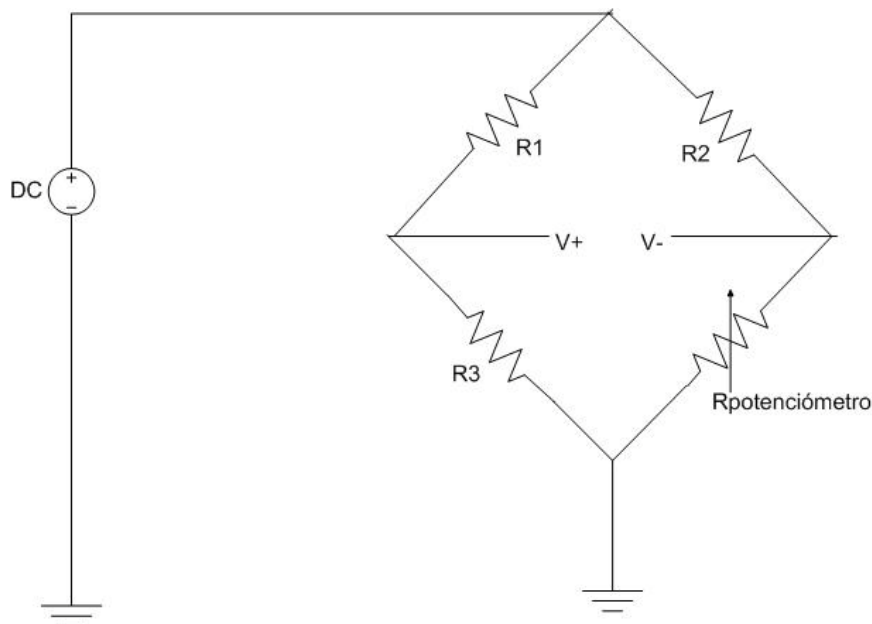
Los potenciómetros son, como se ha dicho, componentes de resistencia variable. La resistencia interna varía en función de una posición que tiene el potenciómetro de forma lineal. Existen tanto potenciómetros que miden la

posición lineal como angular. Los más comunes son los que miden la posición angular.

$$R_{potenci\acute{o}metro} = k * posición \quad (6.1)$$

La **ecuación 6.1** muestra la variación de la resistencia del potenciómetro en función de la posición de este, siendo  $k$  la constante que define la relación.

La idea de funcionamiento sería medir una tensión que va variando de forma lineal con la variación de la resistencia. Para ello, se necesita realizar un acondicionamiento del sistema. Este acondicionamiento se realizaría con un puente de WheatStone (**Figura 6.3**), resultando la tensión de salida a medir que indica la **ecuación 6.2**.



**Figura 6.3:** Acondicionamiento del potenciómetro

$$V^+ - V^- = V_{DC} * \left( \frac{R1}{R3+R1} - \frac{R_{potenci\acute{o}metro}}{R2+R_{potenci\acute{o}metro}} \right) \quad (6.2)$$

El problema de este sistema es que no es lineal. Para conseguir que sea lo más lineal posible, se debe utilizar una  $R2$  mucho mayor que la resistencia del potenciómetro.

### 6.2.2 Motivos del descarte

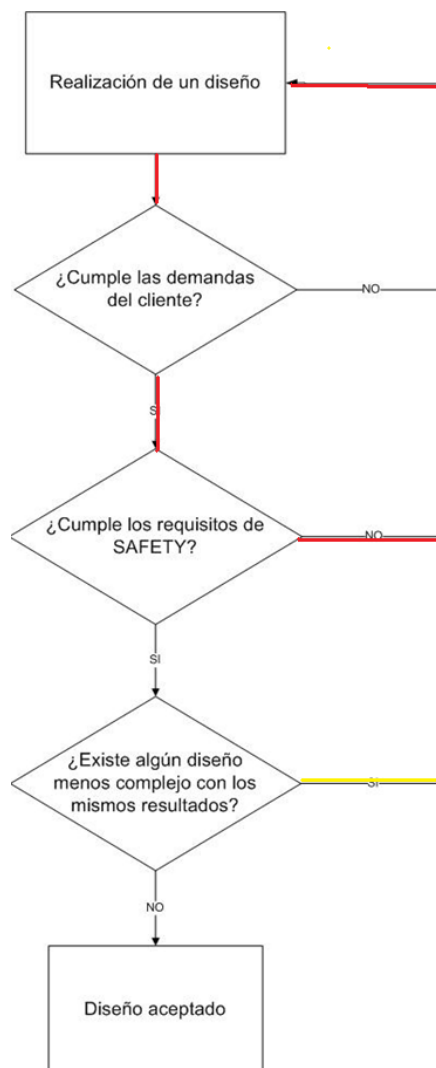
La decisión de excluir cualquiera de estas propuestas se basa en la dificultad que conlleva la colocación física del sensor sobre el servo y las connotaciones de seguridad que ello conlleva. La utilización de cualquiera de los sensores de desplazamiento angular mencionados como el potenciómetro, conlleva una

compleja colocación. Se necesita realizar una conexión física con el servo. El hecho de colocar tal conexión, añade cierta resistencia al movimiento debido a la fricción, con lo que puede afectar al correcto comportamiento mecánico de las superficies. Ello incumple el requisito de seguridad [RS036].

Además, el gasto en tiempo que conlleva la realización de un sistema capaz de ser colocado, no compensa, ya que el proyecto no busca centrarse en un sistema físico para colocar un sensor, sino en la realización de un sistema electrónico para un UAV y su consecuente proceso de seguridad.

Además, conlleva la complicación de tener que alimentar al sensor. La alimentación tendría que estar muy limpia de ruido para que la señal del sensor fuera realista, y por tanto sería necesario preparar un sistema de alimentación muy costoso y complejo para tal acto.

La **Figura 6.4** ilustra el proceso iterativo realizado, marcando en rojo el camino hecho y demostrando que el diseño no era factible. Además, en amarillo está marcado que, si se hubiera llegado a ese punto, tampoco lo hubiera pasado.



**Figura 6.4:** Proceso iterativo realizado con el sensor resistivo

## 6.3 Sensores capacitivos

### 6.3.1 Tecnología del sensor

Se planteó la posibilidad de utilizar sensores capacitivos para la medida de la posición de las superficies. Estos sensores funcionan según la variación de la capacitancia. La capacitancia se rige según la ecuación:

$$C = \epsilon_0 \epsilon_r \frac{A}{d} \quad (6.3)$$

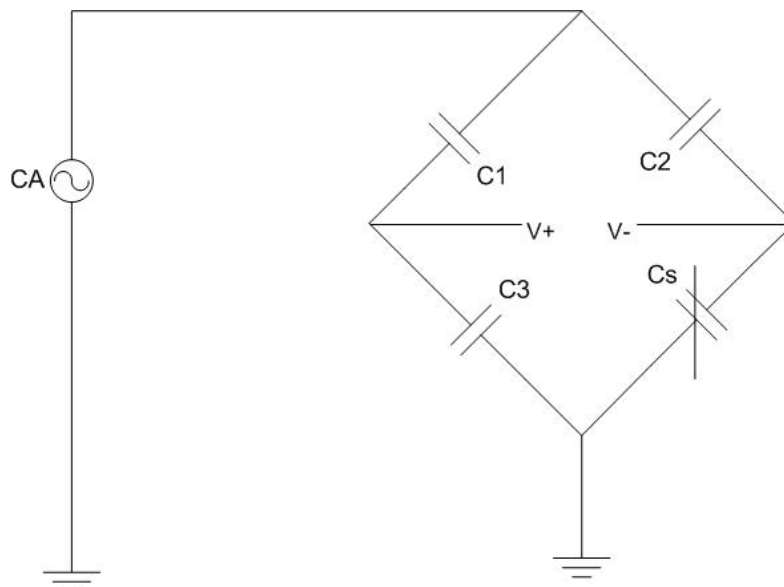
$\epsilon_0$ : constante dieléctrica del vacío

$\epsilon_r$ : constante dieléctrica o permitividad relativa del material dieléctrico de las placas

$A$ : el área efectiva de las placas

$d$ : distancia entre las placas o espesor del dieléctrico

La capacitancia del sensor varía en función únicamente de la distancia entre las placas, ya que tanto el área efectiva de las placas como las constantes dieléctricas no cambian. La adaptación de este sensor se realiza mediante un puente de Wheatstone, tal y como se ve en la **Figura 6.5**. El condensador  $C_s$  sería el sensor capacitivo.

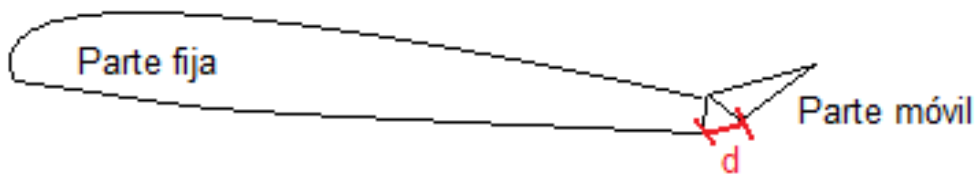


**Figura 6.5:** Adaptación de un sensor capacitivo

La **ecuación 6.4** es la resultante de la adaptación.

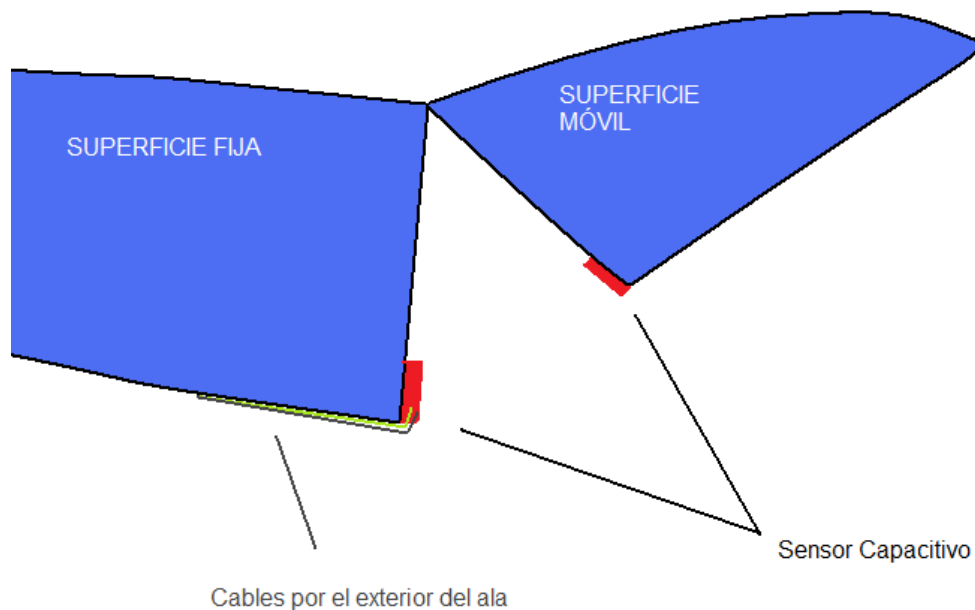
$$V^+ - V^- = V_{CA} * \left( \frac{C_1}{C_1 + C_3} - \frac{C_2}{C_2 + C_s} \right) \quad (6.4)$$

La idea sería utilizar un sensor capacitivo que midiera la distancia que había desde la parte fija del avión a la superficie móvil. En función de la posición de la superficie, la distancia varía y con ello la capacitancia del sensor. Esa variación de la capacitancia es la que se mediría y de ahí se extraería la distancia y por tanto la posición de la superficie. La **Figura 6.6** es un esquema donde se ve la distancia que debería medir el sensor capacitivo.



**Figura 6.6:** Ejemplo de la distancia a medir por un sensor capacitivo

Si se hubiera utilizado este sistema, se habría puesto el sensor unido a la parte fija y una placa metálica en la parte móvil que sería la placa con la que se crearía la capacitancia (y la placa con la que habría que medir la distancia). La **Figura 6.7** muestra un dibujo de la localización del sensor en el ala.



**Figura 6.7:** Dibujo de localización del sensor

### 6.3.2 Motivos del descarte

El mayor problema de la utilización de este diseño proviene de los requisitos de seguridad. En concreto, el requisito de seguridad [RS037] que proviene del PSSA especifica que el sistema no puede modificar la aerodinámica del UAV. Este sistema requiere utilizar conexiones de cables y piezas electrónicas por la superficie del UAV, y por tanto, crea una variación en la aerodinámica de vuelo. El UAV no está preparado para tal acto y por tanto es posible que la creación de este diseño pudiera producir un fallo catastrófico. No solo eso, es posible que las piezas se despeguen y se perdiera el sistema.

Además, la utilización de piezas por la superficie implica un estudio detallado de cómo instalarlo de forma que la fricción con el aire del exterior y las condiciones meteorológicas, así como posibles choques con objetos extraños, no afecten al sistema. Requeriría por tanto un Intrinsic Hazard Analysis, con pruebas tales como mojar constantemente el sistema, lanzar un pájaro contra el avión, etc. Son pruebas que sencillamente no se pueden realizar. Para más información ver [2] y [3]. La complejidad que añade esto al sistema es demasiado alta y no se puede asumir.

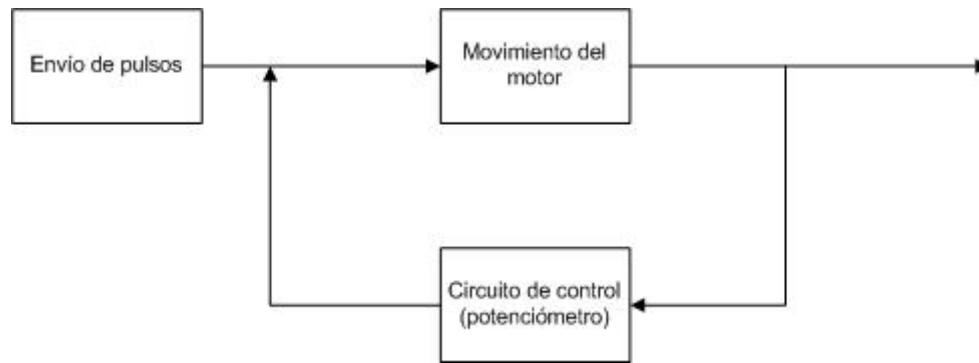
Otro factor que añade complejidad es que se necesita que el sensor esté alimentado por corriente alterna. Todo el sistema de alimentación del UAV funciona con corriente continua. La complejidad se le añade al tener que realizar una conversión DC a AC. Aunque no es el problema principal, es mejor utilizar otras opciones. Uno de los problemas más graves es la aparición de resistencias parásitas, que añaden mucho ruido y falsean la señal recibida. Además, es un sistema que se ve muy fácilmente afectado por campos electromagnéticos. El motor del UAV genera un campo electromagnético muy grande y puede fácilmente afectar al sistema. Sería por tanto necesario realizar un estudio de compatibilidad electromagnética muy elaborado, algo bastante complicado y difícil de realizar en el poco tiempo disponible.

La **Figura 6.4** del apartado anterior también es fácilmente aplicable a este diseño, y por tanto se puede ver como se descartó.

## 6.4 DECISIÓN FINAL: POTENCIÓMETRO DEL SERVO

En el estudio del UAV se estudian las superficies y sus actuadores. En este caso, los actuadores, los servos, son estudiados para conocer su funcionamiento. Durante el estudio del funcionamiento interno del servo, comprobamos que los servos controlan la posición que tienen mediante un circuito de control de lazo cerrado, que controla constantemente que la posición en la que se encuentra es la posición correcta. La **Figura 6.8** muestra un esquema básico del circuito de control.





**Figura 6.8:** Circuito de control de lazo cerrado del servo

El circuito de control utiliza un potenciómetro para conocer por sí mismo la posición del servo. Por tanto, el circuito de control utiliza un sensor del cual se puede extraer la información (la tensión) conectando un sistema que procese los datos en paralelo. Sería como duplicar el circuito de control para mostrar la información del sensor.

Esto conlleva unas ventajas muy importantes ya que simplifican el diseño, pues no es necesario añadir ningún sistema mecánico para los sensores. El procesamiento de los datos se puede realizar fácilmente con cualquier microcontrolador con capacidad para recibir entradas analógicas. En el **ANEXO 2** y el **ANEXO 3** se detalla el funcionamiento de los servos y el potenciómetro interno.

La realización del diseño detallado y como se ha utilizado el potenciómetro de forma más específica, se detalla en el capítulo 8, diseño detallado.

En cuanto a seguridad, las implicaciones que tienen el sistema son muy importantes pues es necesario tocar un circuito de control de un sistema crítico del avión. En el capítulo del Zonal Safety Assessment y en el diseño detallado se concreta como se han solventado los problemas de seguridad del sistema.

## CAPÍTULO 7. ZSA

### 7.1 Introducción

Tomando como referencia el documento de diseño arquitectural se hizo una inspección de aquellas zonas que alojarían nuestro sistema y de cómo la implementación de nuestro sistema en la aeronave podría interferir o alterar su funcionamiento original. Teniendo en cuenta el consumo del hardware, los problemas que se podrían derivar de embarcar el sistema en el UAV y el posible funcionamiento erróneo de nuestro sistema.

Muchas veces los análisis de safety caen en el error de basarse en esquemas rígidos. Estos esquemas muchas veces no tienen en consideración las implicaciones de instalar físicamente el sistema en una aeronave. Debemos tener en cuenta que los diferentes subsistemas que forman parte del avión no son independientes entre ellos. Por ello es necesario hacer un análisis que tenga en cuenta las implicaciones que la instalación de un sistema puntual pueda tener en los sistemas próximos a él. Éste análisis es llamado zonal safety analysis.

En un ZSA debemos analizar zona por zona de la aeronave. La división del avión por partes, por tanto, será una tarea previa a la realización de este análisis. Podemos consultar un ejemplo de división en la **Figura 7.1**. Los resultados extraídos de este análisis serán entradas importantes en la elaboración del SSA y complementarán a sus requisitos.

El objetivo de un Zonal Safety Analysis es asegurar que el diseño del sistema y su instalación cumple con los estándares básicos de diseño y instalación, el efecto de los fallos del sistema en la aeronave, la implicación de los errores de mantenimiento y que cumple con los requisitos de independencia especificados en el FTA.

Durante el proceso de ZSA deberemos evaluar todas las zonas dónde está previsto instalar nuestro sistema, describiendo que encontramos en cada zona y los espacios libres dónde ubicaremos nuestros sistemas. Una vez identificados todos los elementos cercanos deberemos hacer una lista con cada uno de los sistemas que pueden derivar en un malfuncionamiento del sistema y explicar que efectos tendría en el comportamiento de la aeronave. Los resultados de éste ZSA estarán incluidos en el reporte, y en caso de que exista alguna desviación frente al diseño inicial, ésta se reportará al proceso de diseño dónde deberán efectuar los cambios pertinentes.

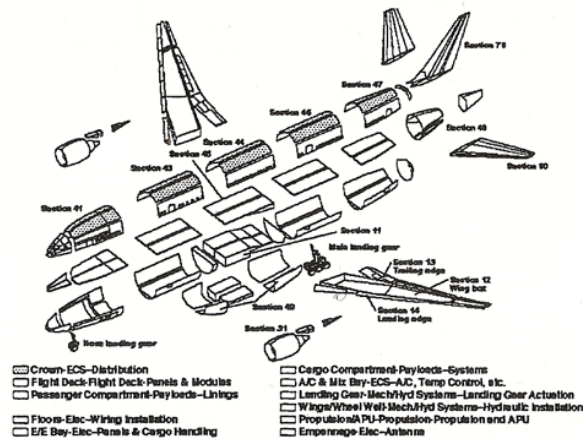


Figura 7.1: Ejemplo división del avión en subsistemas.

## 7.2 Localización de nuestro sistema dentro de la aeronave

Finalmente por cuestiones de tiempo y presupuesto nos hemos limitado a crear un prototipo del sistema a embarcar en las diferentes superficies. Aún y así en este apartado describiremos las zonas en que pensábamos embarcar nuestros sistemas ya que así figura en el ZSA realizado. Nuestro sistema originalmente estaba pensado para implantarse en todas las superficies de control del “Shadow” donde íbamos a colocar un sensor en cada uno de los servos de los dos alerones, uno en el timón de profundidad y otro en el timón de dirección. El cableado desde nuestros sistemas a la estación emisora del Shadow según diseño se pasaría por los mismos conductos que utiliza el cableado del sistema de control. En la **Figura 7.2**, proporcionada por el departamento de diseño, podemos observar el recorrido del cableado de control y la ubicación de nuestros sistemas.

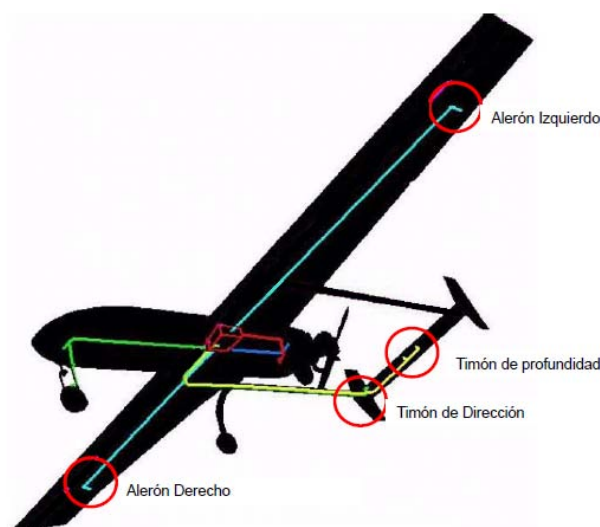
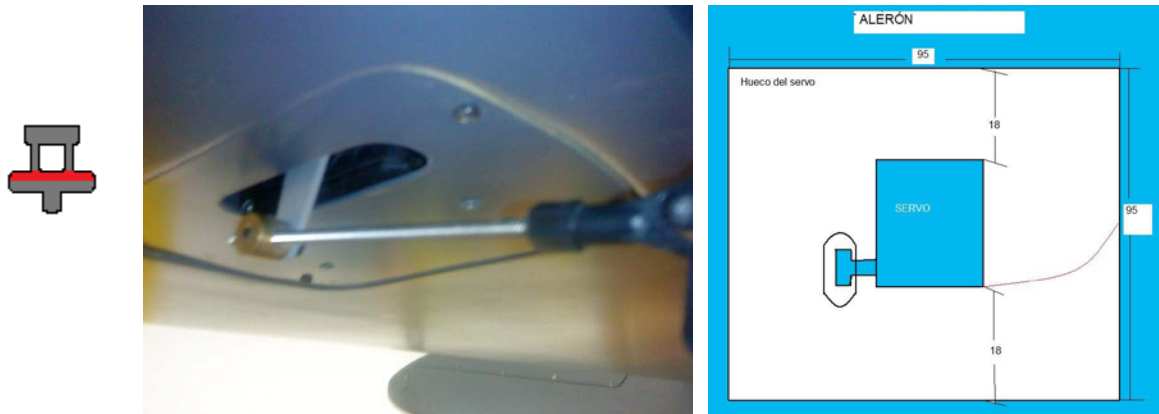


Figura 7.2: Ubicación de nuestro sistema y cableado de control

### 7.2.1 Alerones

Según la última revisión del documento de diseño el sensor de los alerones se iba a alojar en el interior del ala ya que ésta está completamente hueca como podemos observar en la **Figura 7.3**. Entonces desde el departamento encargado de llevar a cabo el proceso de ZSA se analizó la zona y se determinó que en la zona escogida por el departamento de diseño íbamos a encontrar a más del servomotor del que tomar la medida de voltaje, el cableado de alimentación de éste y las líneas de comunicación del sistema de control.

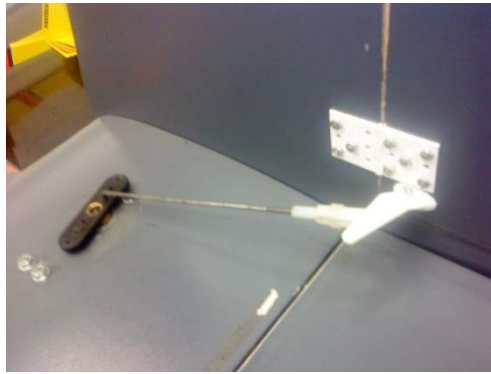
Desde el departamento de safety se reportó a diseño la conveniencia de instalar nuestro sistema lo más lejos posible de las líneas de control y buscar métodos para asegurar la correcta fijación del sistema a la superficie del ala.



**Figura 7.3:** Servomotor del alerón izquierdo

### 7.2.2 Timón de dirección

Según diseño, el sensor del timón de dirección se alojará (en caso de ser implementado nuestro prototipo en el futuro) en el interior de la tubería que une las alas con las superficies de control de guiñada y picado, podemos observar esta cavidad en la **Figura 7.4**. En el proceso de ZSA se identificó que por la zona escogida pasan los conductos de control y el cableado de alimentación de los servomotores.

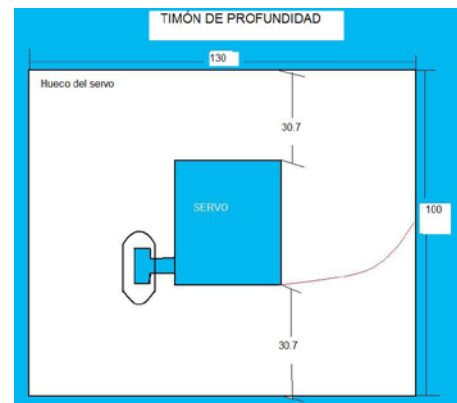


**Figura 7.4:** Cavity en la que instalar nuestro sistema.

### 7.2.3 Timón de profundidad

Según la última revisión del documento de diseño el sensor del timón de dirección se iba a alojar en el interior de éste ya que esa zona está completamente hueca. Con esto, durante el proceso de ZSA se encontraron en la zona escogida por el departamento de diseño, además del servomotor del que tomaremos la medida (**Figura 7.5**); el cableado de alimentación de éste y las líneas de comunicación del sistema de control.

Desde el departamento de safety, al igual que en el caso de los alerones, recomendamos el instalar nuestro sistema lo más lejos posible de las líneas de control y buscar métodos para asegurar la correcta fijación del sistema a la superficie del ala.



**Figura 7.5:** Servomotores del timón de profundidad.

## 7.3 Identificación de todos los subsistemas que sean un problema en potencia.

### 7.3.1 Potenciómetro

Al soldar dos cables al potenciómetro para tomar una medida de la resistencia de éste, estamos tocando un subsistema perteneciente a la función de control de la aeronave, que como podemos ver en la **Figura 7.6** es crítica para el funcionamiento del avión. Por lo que según el documento [6] nuestro sistema pasa de tener una “major condition failure” a “catastrophic condition failure”. Esto ahora implica que todas aquellas funciones de nuestro sistema (y todos los subsistemas que forman parte de ellas) que puedan interferir en el correcto funcionamiento del potenciómetro pasan a tener el mismo DAL que el servomotor. En nuestro caso se trata de un DAL A al no haber servomotores redundantes. Por este motivo algunos requisitos de safety se actualizarán para cumplir con esta nueva condición. La máxima prioridad ahora será no falsear la medida que el sistema de control propio del servomotor toma del potenciómetro.

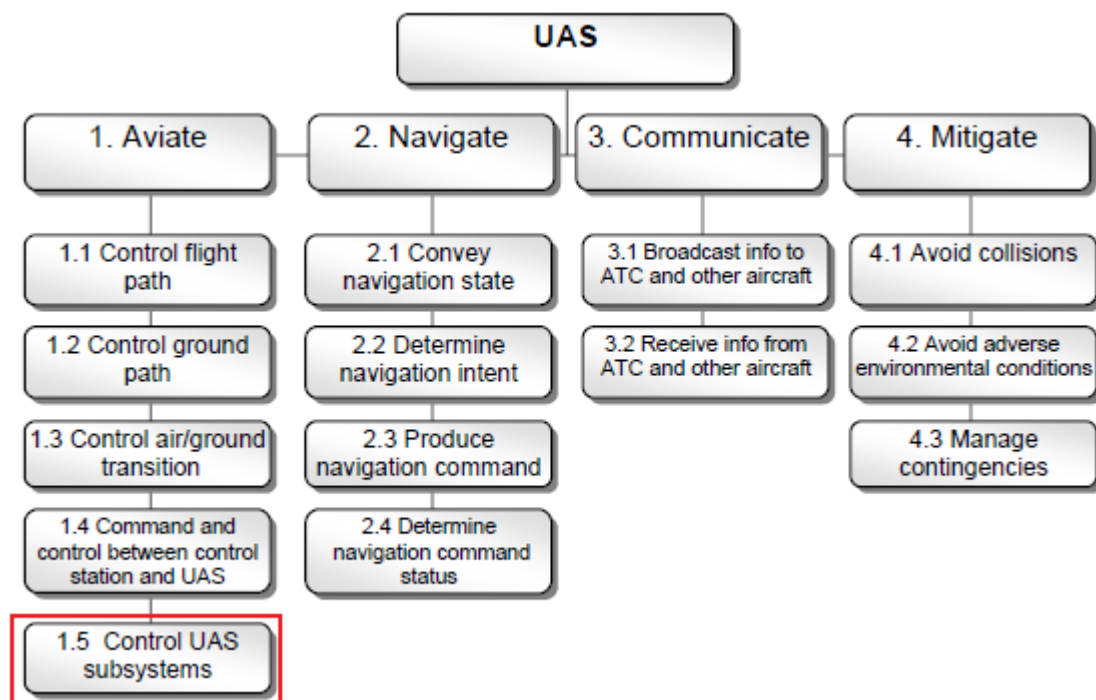


Figura 7.6: Funciones de alto nivel de un UAS

Deberemos trazar al proceso de diseño todos los requisitos generados para evitar falsear la medida que el sistema de control propio del servomotor toma del potenciómetro así como la actualización de los requisitos derivados de la actualización del DAL a satisfacer. Diseño deberá revisar si cumple con lo expuesto en los nuevos requisitos y si no deberá ajustarse a estos realizando

los cambios convenientes. Recordar que estamos dentro de un proceso iterativo.

### **7.3.2 Encapsulado del servomotor.**

Tal y como figura en el diseño deberemos hacer pasar cableado desde el interior del servo a su exterior, por esto deberemos perforar el encapsulado. Esto según el proceso de ZSA reportó, podría tener consecuencias en el aislamiento del servo (temperatura, agentes climáticos, pequeña fauna, etc.), comprometiendo así su seguridad.

Para asegurar que desde diseño se tenían en cuenta estas consecuencias al perforar el encapsulado del servomotor se reportaron los correspondientes requisitos.

### **7.3.3 Encapsulado de nuestro sistema (no puesto en práctica)**

Este es uno de los apartados que no hemos implementado en el prototipo. Aún así si que lo hemos contemplado en el ZSA. Las conclusiones de este proceso reportaron que el encapsulado de nuestro sistema tendrá que adecuarse a los espacios que nos brinda la arquitectura del UAV y primando en todo momento que éste no interfiera con los sistemas mecánicos de la aeronave. Los detalles de estos espacios están explicados en el capítulo 2 del ZSA adjunto. Destacar también que de éste apartado se reportaron dos requisitos a diseño que no pudieron cumplirse al final.

### **7.3.4 Malfuncionamiento general del sistema**

Debido a que nuestro sistema se embarcará en una aeronave debemos tener en cuenta como un funcionamiento erróneo de éste podría afectar a la resta de sistemas del avión. En el proceso de ZSA hemos visto como nuestro sistema estará en contacto con cableado del sistema de control y alimentación de los servomotores. Por este motivo deberemos considerar situaciones límite como puede ser un sobrecalentamiento del sistema. También es importante estudiar un posible desprendimiento de nuestro sistema dentro del cuerpo de la aeronave, lo que sin duda afectaría en gran medida a las actuaciones mecánicas de nuestro avión.

Al contemplar que el sobrecalentamiento o desprendimiento de nuestro sistema puede comprometer la integridad del avión, consideremos que el encapsulado de nuestro sistema tendrá un DAL igual al del servomotor. En nuestro caso se trata de un DAL A al no haber servomotores redundantes.

## 7.4 Como evitaremos los anteriores problemas potenciales.

### 7.4.1 Potenciómetro

Para evitar falsear la medida que el sistema de control recibe del potenciómetro colocaremos nuestro sistema en paralelo, asegurándonos mantener la caída de tensión entre los bornes. Además, se conectará a un amplificador operacional no inversor, para que la corriente de entrada a nuestro sistema sea igual a 0 y por tanto toda la corriente se quede en el servo.

### 7.4.2 Encapsulado del servomotor

Según la última revisión del documento de diseño, para asegurar que nuestro sistema no compromete en nada la integridad de la estructura interna del servomotor una vez pasado el cableado por la carcasa del servo sellaremos el agujero con una gota de silicona haciendo presión entre las dos partes que conforman la carcasa de nuestro servo. El resultado una vez terminado el prototipo es como el que podemos observar en la **Figura 7.7**.



**Figura 7.7:** Cableado a extraer del servo.

### 7.4.3 Encapsulado de nuestro sistema (no implementado)

Este apartado no se ha contemplado en el diseño final del prototipo ya que entendemos que al no estar embarcado y ser solo una versión de demostración no era algo crítico. Desde el punto de vista de safety esto comporta que una serie de requisitos no se han cumplido, pero debemos mencionar que el estudio de safety se desarrolló en base a la instalación de nuestro sistema sensor en las cuatro superficies que anteriormente hemos mencionado.



#### **7.4.4 Malfuncionamiento general del sistema**

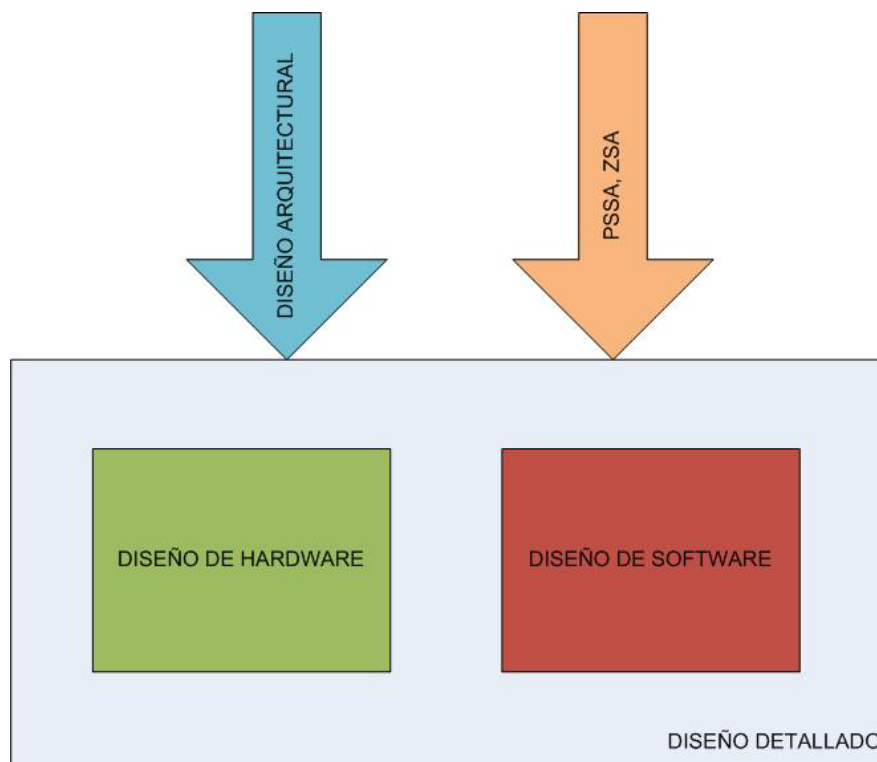
Desde el ZSA se reportaron a diseño los requisitos que debía cumplir el sistema para ser tolerante a sobrecalentamientos y desprendimientos en el interior de la aeronave. Así mismo se destacó que la probabilidad de que se llegue a dar un funcionamiento anómalo tan catastrófico como para dañar los sistemas colindantes al nuestro es casi inexistente. Desde diseño al centrarnos finalmente en un prototipado del sistema no se ha dado solución a estos requisitos al no considerar su embarcado en el avión. En fases previas del diseño dónde esta instalación dentro de la aeronave si fue se estimó positivo el uso de un encapsulado térmicamente aislante y autoextingible así como la fijación de nuestro sistema con silicona resistente a los cambios de temperatura.

## CAPÍTULO 8. DISEÑO DETALLADO

### 8.1 Introducción

El diseño detallado es el proceso en el cual se define y se detalla todo el sistema. Es el momento de tomar las decisiones de que piezas se utilizarán, como se va a controlar el software, etc. El ANEXO 3, Diseño detallado, contiene toda la información sobre el diseño detallado del sistema, incluyendo las diferentes partes y piezas, y como se implementara el software.

El desarrollo del diseño detallado, se dividió en dos bloques principales: diseño del hardware y diseño del software. Lógicamente, el diseño detallado viene condicionado por todos los requisitos de diseño provenientes del diseño arquitectural, y totalmente unido a los informes de seguridad desarrollados. La **Figura 8.1** muestra un esquema de esta división y la procedencia de los datos para el diseño.



**Figura 8.1:** División del diseño detallado y procedencia

Un aspecto muy importante es saber que, para el diseño detallado, se ha centrado la atención en realizar un sistema que pueda utilizarse en cualquier superficie o sistema que utilice un servo como actuador. Así, no se ha realizado un sistema completo para el UAV, sino que se ha desarrollado un sistema fácilmente reproducible y que se podría aplicar tanto en el UAV como en otro sistema con servos. No obstante, se han seguido todos los requisitos del sistema, incluidos los de seguridad, para poder asegurar que el sistema sería embarcable en el UAV.

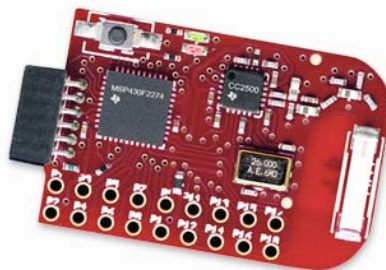
## 8.2 Diseño del Hardware

Tal y como se ha explicado en el párrafo 6.4, el sensor elegido para el sistema es el potenciómetro que lleva incluido el propio servo. Así, la parte de hardware se concentra en cómo realizar la conexión entre el potenciómetro y el microprocesador, y el envío de este.

Por tanto, lo más importante es que el sistema sea capaz de coger la señal analógica (la tensión) que sale del potenciómetro, y tratarla para poder mandarla a través de la red de comunicaciones del UAV, en este caso la red MAREA.

Para ello, es necesario utilizar un conversor A/D de forma que el microcontrolador que vaya a tratar la información pueda procesarla (puesto que funcionan con señales digitales. Tras la recepción, el microcontrolador procesará la señal y la emitirá a través de la red Ethernet, y a través de una conexión serie, tal como especifican los requisitos.

Debido a lo avanzado de la tecnología actual, se busca un microcontrolador que ya incorpore el conversor A/D. Con los cálculos realizados, se necesita un conversor A/D de mínimo 5 bits. Se decide utilizar así, un módulo conocido (MSP430 RF2500, **Figura 8.2**) que incorpora el microcontrolador MSP430F2274. Este micro tiene integrado dos conversores A/D, entre otras muchas funciones como amplificadores operacionales e interfaces de comunicación para usar con otros dispositivos.



**Figura 8.2:** Módulo MSP430 RF2500 con el microcontrolador MSP430F2274

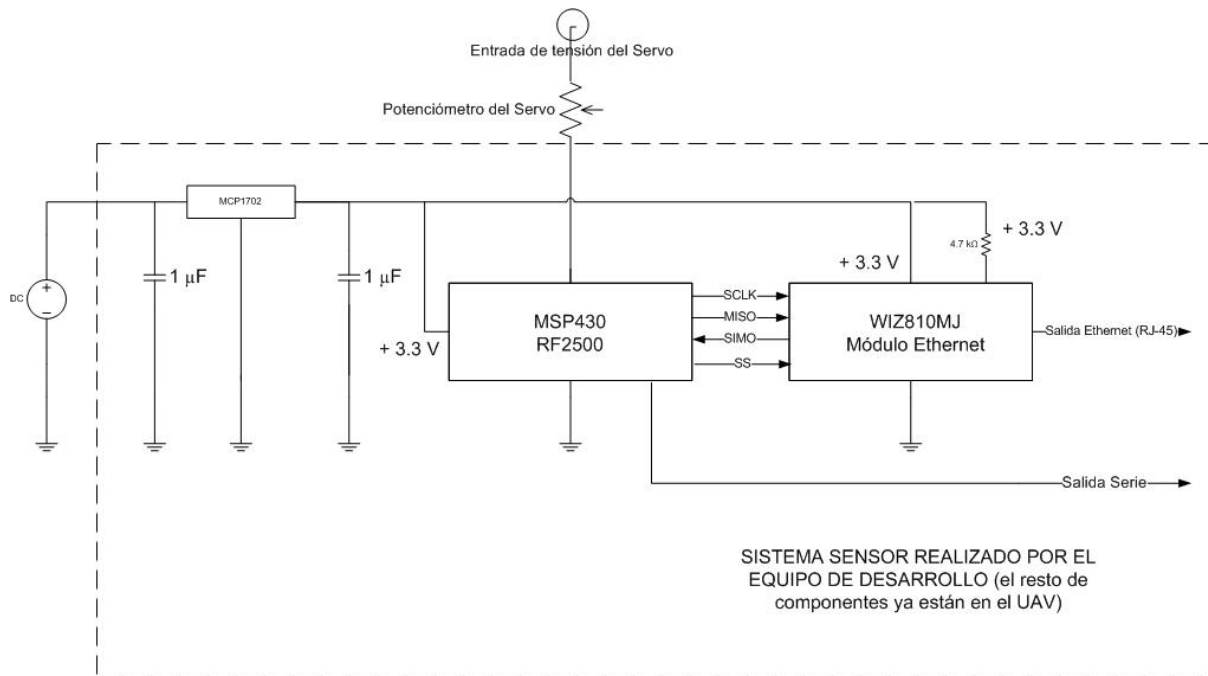
El modulo no puede transmitir por Ethernet, por lo que será necesario encontrar una solución. La mejor es la utilización del módulo de Wiznet WIZ810MJ (**Figura 8.3**), que viene con diversos protocolos de Ethernet incorporados en el Hardware y solo necesita programarse por software los sockets de transmisión. Además, puede funcionar de forma esclava con el microcontrolador gracias a la interfaz de comunicación de este.



**Figura 8.3:** Módulo de Ethernet WIZ810MJ

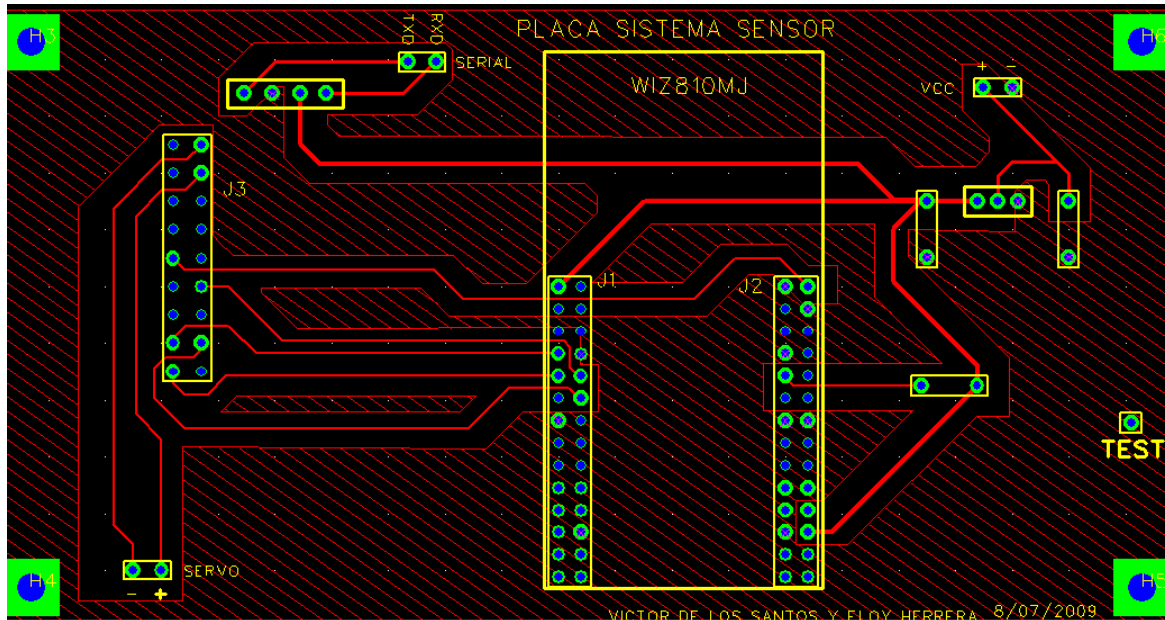
Finalmente, es importante la alimentación del sistema. Todos los componentes necesitan una tensión de entrada de 3.3 V, y por ello se decide utilizar un regulador de 3.3V. La tensión, proveniente de baterías de 5 V, se ve muy afectada por el ruido y llega muy atenuada, poniéndose cerca de los 3.3 V en la entrada del sistema. Para evitar problemas, se decide utilizar un regulador LDO (Low Drop Out) que permite operar con diferencias de tensión muy pequeñas entre la entrada y la salida. Los reguladores necesitan tener en cuenta que deben adaptarse según indique su datasheet.

Así, el esquema de conexiones del sistema se puede ver en la **Figura 8.4**.



**Figura 8.4:** Diseño del sistema sensor

Para poder realizar todas estas conexiones, se desarrolla una placa de circuito impreso o PCB, de forma que se puedan realizar las conexiones fácilmente y de una forma más robusta que si todo fuera formado por cables. El diseño de la placa, realizada con el programa *FREEPCB*, está pensada como prototipo, de forma que no se ha adecuado a cada superficie, y sus medidas pueden cambiar. El diseño de la PCB puede verse en la **Figura 8.5**.



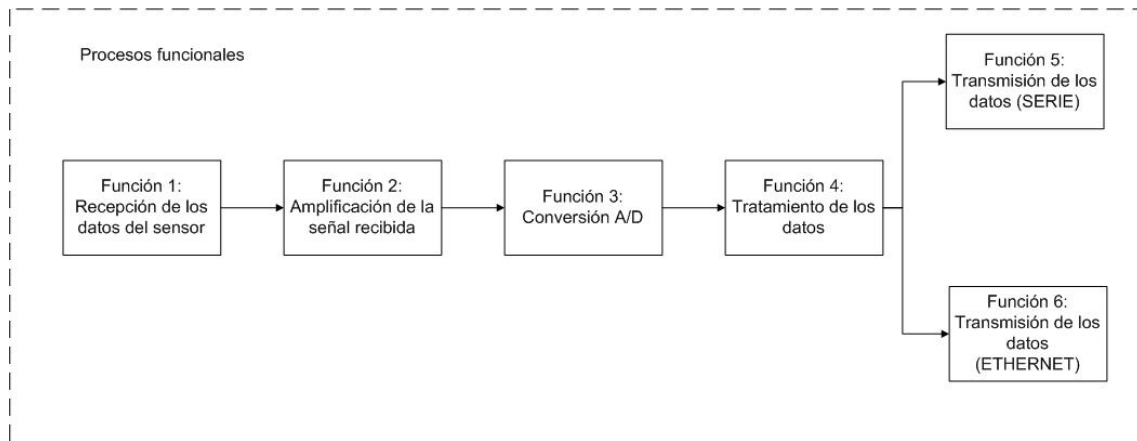
**Figura 8.5:** Diseño de la PCB del sistema

### 8.3 Diseño del Software

El diseño de software es muy importante ya que es el que realizará todo el control del sistema. Es importante realizar un diseño antes de empezar a escribir líneas de código, pues de esta forma el código del software se realiza de forma más ordenada y se puede trazar más fácilmente a los requisitos del sistema.

Así, se decidió dividir el software en 6 funciones principales, para controlar fácilmente cada proceso de éste. La división y asignación de cada función puede verse en la **Figura 8.6**. Esta división queda reflejada en el código mediante comentarios que indican cada función.

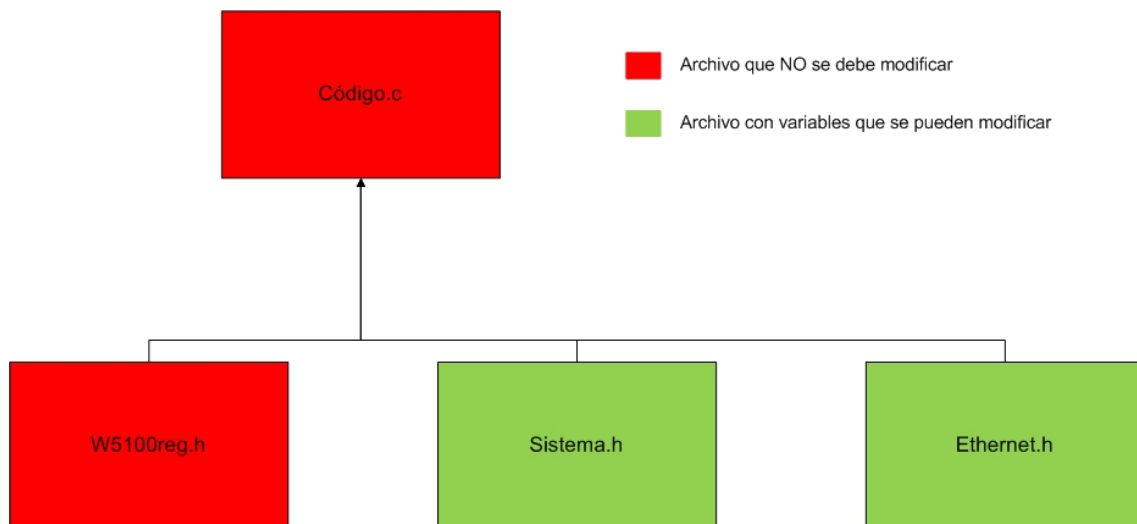
De esas 6 funciones, todas menos la función 4 tienen funciones de configuración del sistema (algunas también tienen además funciones de software). Esas funciones son las funciones denominadas de firmware, que sirven para definir el funcionamiento del microcontrolador y su configuración interna.



**Figura 8.6:** División en funciones del software

El hecho de realizar un sistema que pueda utilizarse en diferentes superficies obliga a que haya que hacer una serie de modificaciones en el software en función de la superficie. Esas modificaciones consisten en simples variables que definen la superficie a medir, incluyendo un modificador. Además, el sistema se tiene que poder adaptar fácilmente a la red MAREA (ver ANEXO 2: Diseño Arquitectural) de la que formaría parte.

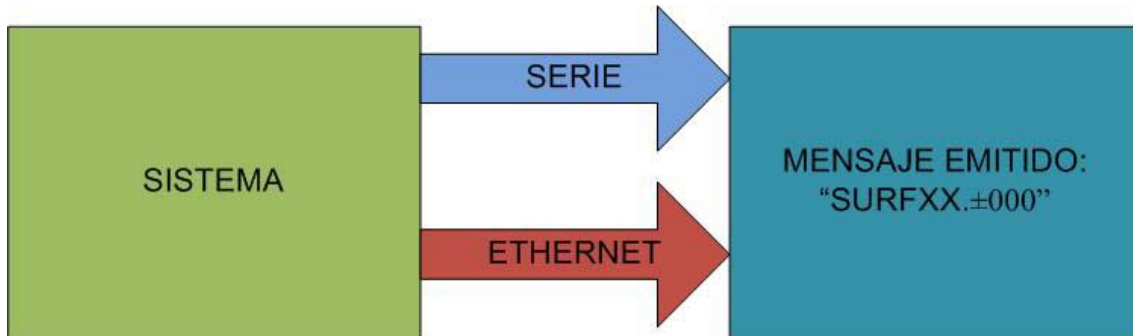
Debido a esto, se realiza una división de archivos que contienen el software, 3 de cabeceras donde van las variables a modificar y el archivo de código, que no se debe modificar. La estructura de ficheros queda representada en la **Figura 8.7**. Todo el código del firmware también está incluido en el archivo código.c, que contiene el código de funcionamiento del sistema. El código puede verse en el ANEXO 12.



**Figura 8.7:** División del software en ficheros

## 8.4 Interfaz

La interfaz consiste simplemente en un mensaje emitido a través del puerto serie y la conexión ETHERNET. Este mensaje, proveniente del sistema, contiene toda la información que queremos conocer. La **Figura 8.8** muestra un esquema de la interfaz.



**Figura 8.8:** Interfaz del sistema

Es un mensaje que incluye la cabecera SURF (de *surface*) para identificar que lo que se emite es la posición de una superficie, seguida del identificador, formado por dos dígitos que identifican a la aeronave. Finalmente, y separados por un punto, se emite la información de la posición. Por ejemplo, si el mensaje proviene de la superficie 07, y está en la posición  $-5^{\circ}$ , el mensaje sería:

"SURF07.-005"

## 8.5 Consideraciones de seguridad durante el diseño detallado

Tal y como está explicado en los capítulos y anexos de seguridad, especialmente en el ZSA, la decisión de utilizar el potenciómetro del servo como sensor propio es una decisión muy arriesgada y debe ser muy estudiada, debido a que se está tomando una señal de control de una superficie de control. Las superficies de control son críticas para el correcto funcionamiento del UAV, y por tanto la conexión de nuestro sistema a la superficie puede provocar un problema que acabe en la destrucción del UAV.

Al conectarnos en paralelo al sistema de control del servo podemos falsear la señal variando la función de transferencia del sistema. Eso se debe a que al conectar en paralelo el sistema, la tensión que captamos y que queda en el circuito de control es la misma y no se ve afectada, pero parte de la corriente que pasa por el potenciómetro, en vez de llegar al circuito de control va a parar a nuestro sistema.

Por eso, se decide utilizar un amplificador operacional. Los amplificadores operacionales están hechos de forma que la impedancia a la entrada sea muy grande, haciendo que la corriente que entre en él sea idealmente cero. Por tanto, toda la corriente se queda en el servo y no afectamos a la función de transferencia del sistema.

Para ello, en vez de añadir un módulo de un amplificador operacional, aprovechamos que el microcontrolador del sistema tiene incorporado uno de forma interna. Así, nos ahorramos tener que añadir uno al sistema y podemos controlarlo por software.

Esto puede suponer un grave problema ya que el microcontrolador puede configurar el puerto al que está conectado el sensor como una salida en vez de como una entrada. Por defecto, la configuración de todos los puertos del microcontrolador es de entrada, y por software está configurada como tal. Pero existe la posibilidad muy remota que debido a causas externas como un problema de EMC, una tormenta solar o cualquier cosa que pueda afectar a los sistemas electrónicos, cambiando esa entrada por una salida. Para poder evitar esto, debería realizarse un encapsulado del sistema metálico capaz de resistir todos estos problemas. Sin embargo, no se ha desarrollado ninguno para centrar la atención en otros aspectos más importantes, ya que es muy remota la posibilidad de que ocurra lo expuesto.



## CAPÍTULO 9. VERIFICACIÓN

### 9.1 Introducción

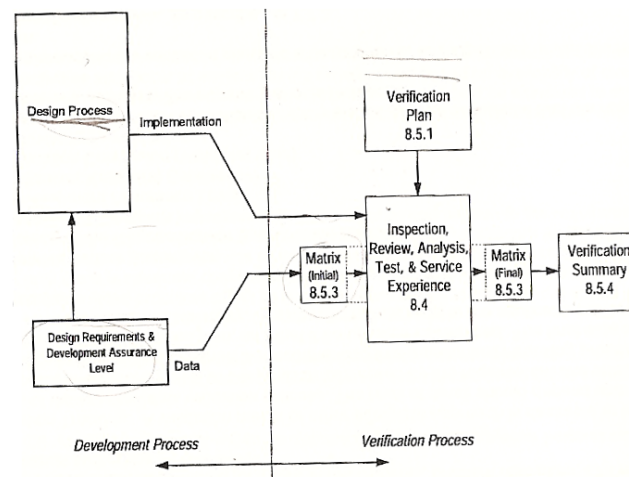
Según el documento [3] el cual hemos seguido como guía para la elaboración de este resumen de verificación, el propósito de la verificación es asegurar que todos los niveles de implementación cumplen con sus requisitos especificados. La verificación consiste en inspecciones, análisis, revisiones, tests i experiencia en servicio acorde a un plan de verificación.

Los procesos de verificación tratan de asegurar que todas las funciones han sido correctamente implementadas, que todos los requisitos han sido satisfechos y asegura que el análisis de safety sigue siendo válido para el sistema implementado.

El proceso de elaboración se compone de tres actividades distintas como podemos ver en la **Figura 9.1**:

- a) **Planning**: Planear los documentos que necesitaremos, la secuencia de actividades a realizar, los documentos que deberemos producir, la recopilación de la información necesaria , selección de actividades específicas y sus criterios de evaluación, y la generación de un hardware y software verificado.
- b) **Actividades o pruebas de verificación**: Las pruebas de verificación son completamente dependientes una de ellas, tanto es así que no se procederá a realizar una prueba nueva hasta que la anterior cumpla con los parámetros de corrección establecidos. Para realizar estas pruebas nos basaremos en tres métodos principales:
  - **Inspección y revisión**: Se consensuará si el sistema cumple con sus requisitos tales como su implementación física, si su diseño se corresponde a lo descrito en el documento de diseño y si cada uno de los elementos que componen el sistema cumplen a la vez con sus requisitos. Para éste tipo de tareas nos ayudaremos con unos checklist redactados por el equipo de verificación.
  - **Testeado**: Mediante el procedimiento de test evidenciaremos la repetida corrección en el tiempo de nuestro sistema. Demostrando así el cumplimiento de los requisitos. Usaremos un estricto criterio de Válido/Fallido. Para la realización de este tipo de pruebas deberemos consultar los documentos de test descritos por el equipo de verificación.
  - **Parecido/Experiencia**: Realmente la verificación del sistema total se basará en aportar evidencias de la corrección que nuestro sistema podrá tener al ser implementado en las diferentes superficies del Shadow o cualquier otra aeronave.

- c) **Documentación:** Como documentos de verificación facilitaremos los checklist y documentos de test descritos anteriormente así como una matriz de verificación donde se incluirán los requisitos a verificar, la función asociada al requisito, en DAL (Development Assurance Level), los métodos aplicados en la verificación, la conclusión: válido/fallido y un resumen dónde anotar posibles eventualidades.



**Figura 9.1:** Proceso de verificación

## 9.2 Matriz de verificación

Una matriz de verificación (o verification matrix) u otro documento de guía equivalente deberán implementarse para seguir el proceso de verificación. El nivel de esta matriz suele depender del DAL del elemento o sistema a verificar. Estas matrices pueden tener el formato que el departamento de verificación elija, pero deben contener como mínimo:

- Los requisitos.
- Función asociada al requisito.
- DAL.
- Método aplicado para verificar el requisito.
- Conclusión (Pasado o fallido)
- Eventualidades ocurridas durante el proceso de verificación.

Los métodos aplicados para verificar un requisito podrán ser cualquiera de los mencionados anteriormente:

- a) **Inspección y revisión:** Consultar anexo (**ANEXO 10**). Este anexo es una lista con diferentes aspectos sencillos a inspeccionar, usaremos un criterio de pasado o fallido en función si se ha podido observar que se cumple aquello enunciado por el checklist o no.

- b) **Testeado:** Consultar el **ANEXO 10**. Estos contienen unos procedimientos de test con unos pasos muy claros y sencillos a seguir y al final de los cuales deberemos decidir si el sistema pasa el test o no acorde a unas tolerancias dadas. Los tests tendrán la siguiente estructura:
- Item testeado.
  - Equipo necesario para realizar el test.
  - Procedimiento del test
  - Tolerancias aceptadas.
- c) **Parecido/Experiencia:** Realmente la verificación del sistema total se basará en aportar evidencias de la corrección que nuestro sistema podrá tener al ser implementado en las diferentes superficies del Shadow o cualquier otra aeronave. No podemos establecer un ciclo acelerado en una cámara anicónica dónde probar que nuestro sistema funciona sin ningún incidente durante las horas que sería necesario para establecer que cumplimos un DAL A.

En la **Tabla 9.1** podemos ver un extracto de la matriz de verificación que encontraremos en el **ANEXO 10**.

**Tabla 9.1:** Extracto de la matriz de verificación

Requisito	Función asociada	DAL	Método de Verificación	Conclusión	Eventualidades
[R001]	Medir actuaciones timón de profundidad.	C	Test 01		
[R002]	Medir actuaciones timón de dirección.	C	Test 01		
...	...	...	...	...	...

### 9.3 Reporte de problemas

Otra de las partes que debe contener un informe de verificación es un reporte con todos los problemas que nos hayamos encontrado durante el proceso de verificación. Cuando un test tenga un resultado negativo (Fallido), el error que lo haya ocasionado será reportado al departamento de diseño dónde deberán encontrarle una solución.

Los errores se reportarán con el formato de la **Tabla 9.2**:

**Tabla 9.2:** Ejemplo de reporte de problema

Requisito:	Fecha problema:	
	Fecha solución:	
Descripción		
Implicación de seguridad:		

Los parámetros a incluir en este reporte son:

- **Requisito:** Identificador y explicación del requisito de diseño o safety que no se ha conseguido superar
- **Fecha problema:** Día en el que nos hemos encontrado este fallo.
- **Fecha solución:** Día en el que el departamento de diseño nos comunica mediante informe que el problema ha sido solucionado.  
 IMPORTANTE: NO es el día que el departamento de verificación constatar que efectivamente el error ha sido superado.  
 Un recuadro donde el departamento de diseño incluirá una “V” de válido cuando haya podido constatar que efectivamente el error ha sido superado.
- **Descripción:** Describiremos el problema hallado indicando el momento en el que fuimos conscientes de éste y en qué consiste. Aportaremos a más los valores obtenidos y los que se esperaban obtener para facilitar en medida de lo posible la faena del equipo de diseño.
- **Implicación de seguridad:** Las implicaciones que este error tendría en el global del sistema y en la aeronave en caso de no haber sido corregirse.

Para consultar los errores reportados a diseño desde verificación consultar el **ANEXO 10**.

## 9.4 Recopilación de documentos concernientes a la verificación

Por último, para completar el verification summary se nos pide que hagamos un recopilatorio con toda aquella información fuente que hemos utilizado para la redacción de éste y de todos aquellos documentos que hayamos producido.

En la preparación de las pruebas de verificación, que empiezan ya a definirse al empezar el diseño arquitectural, se utilizan gran cantidad de documentos que anteriormente hayamos generado. En nuestro caso estos documentos quedan recogidos en una tabla que puede consultarse en el anexo correspondiente al Resumen de verificación.

Es cierto también que en el proceso de verificación se produce gran cantidad de documentación, la mayoría de ella relacionada con las pruebas a las que debemos someter tanto a hardware como software para verificar su validez y

su corrección respecto a los requisitos establecidos. Los documentos que nosotros hemos generado en la elaboración de este reporte o resumen están citados en una tabla que puede consultarse en el anexo correspondiente al Resumen de verificación

## **9.5 Trazabilidad**

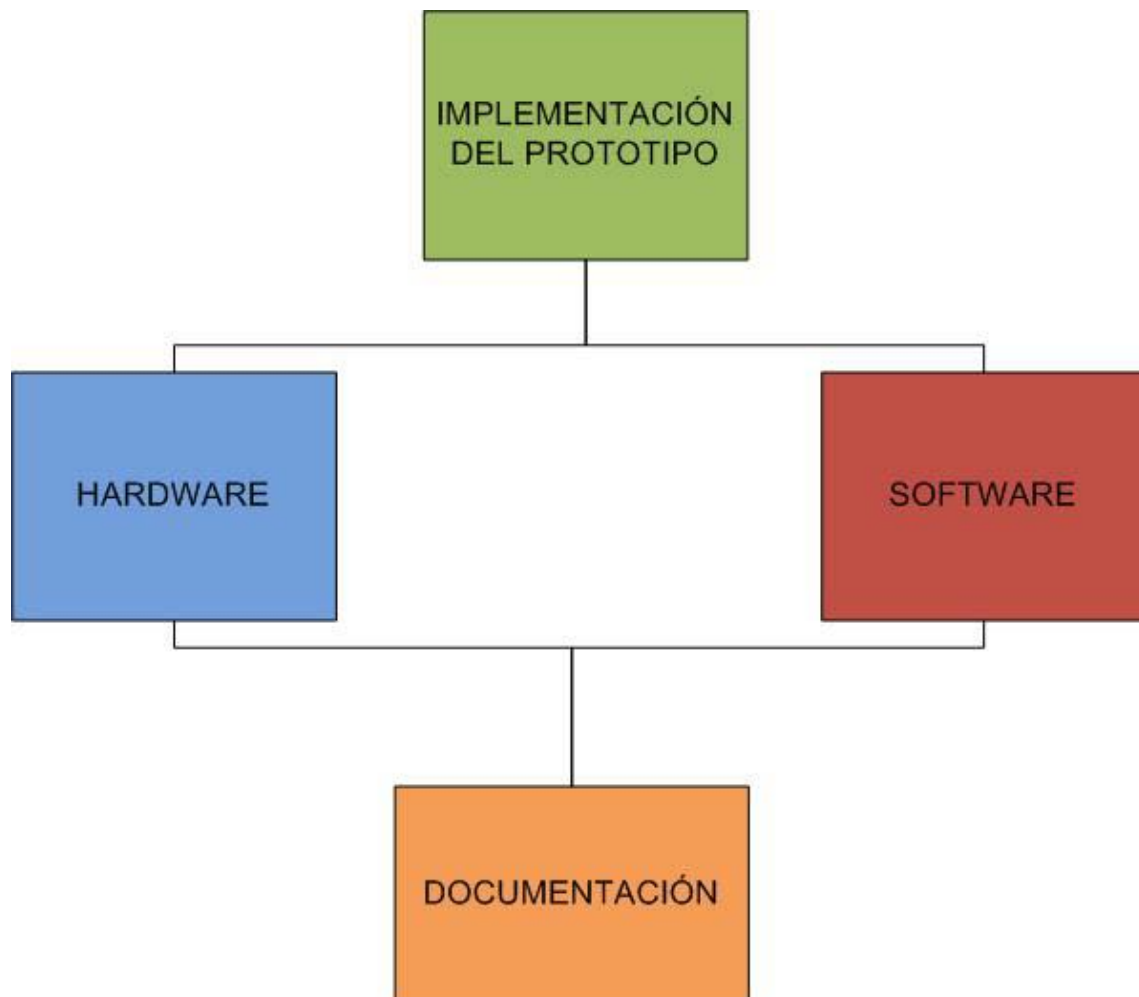
La trazabilidad entre los procesos de verificación y los de diseño es constante. El proceso de diseño debe conocer en todo momento cual es el proceso de las pruebas de verificación del sistema. En el caso de que un requisito no se cumpla o no dé unos resultados dentro del margen de tolerancias aceptadas deberemos enviar enseguida un informe a diseño.

En este informe se especificará el requisito testado y se adjuntarán las pruebas realizadas junto a los resultados obtenidos.

## CAPÍTULO 10. IMPLEMENTACIÓN

### 10.1 Introducción

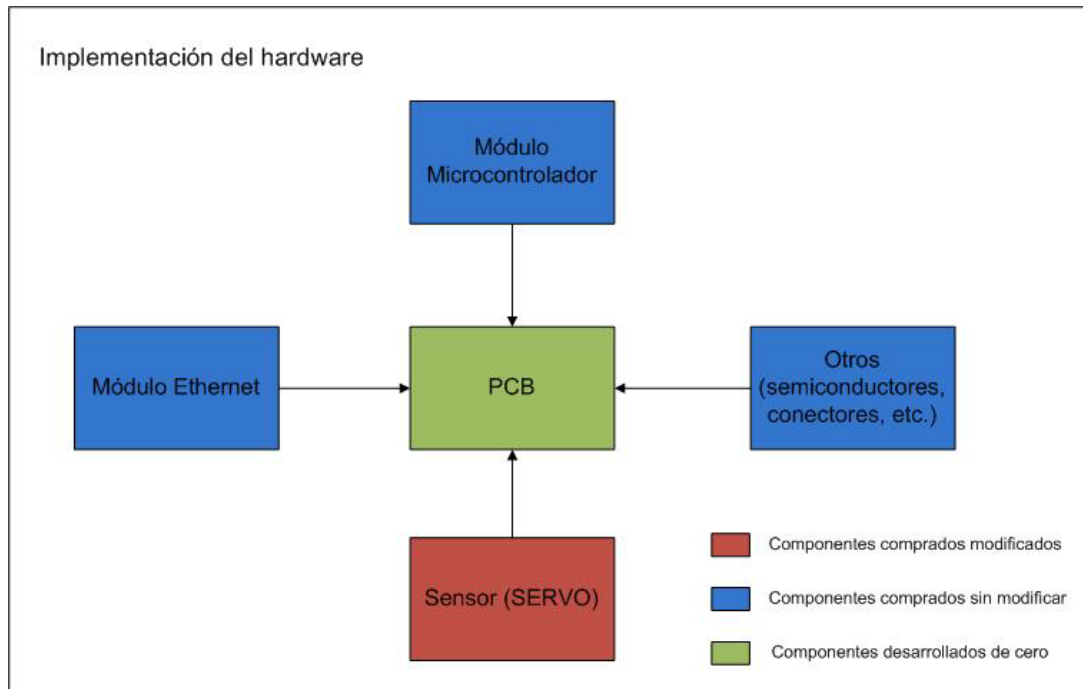
Finalmente, se lleva a cabo la realización de un prototipo para poder comprobar los resultados del sistema. Este prototipo, tal como se ha comentado en apartados anteriores, no es un modelo para implantar ya en las superficies del UAV, sino un modelo inicial que puede sufrir cambios, especialmente en el montaje del hardware y su PCB. En este capítulo final se mostrarán los resultados del desarrollo del sistema. La **Figura 10.1** muestra la división de la implementación en el hardware y el software y como de esta implementación deriva una documentación del sistema.



**Figura 10.1:** Diagrama de implementación del sistema

## 10.2 Resultado del hardware

El hardware se compone principalmente de una serie de módulos y componentes que se sueldan sobre una PCB, tal y como está detallado en el ANEXO 3: Diseño detallado. En la **Figura 10.2** se ve un esquema de los diferentes componentes del software remarcando si se han comprado o se han desarrollado de cero.



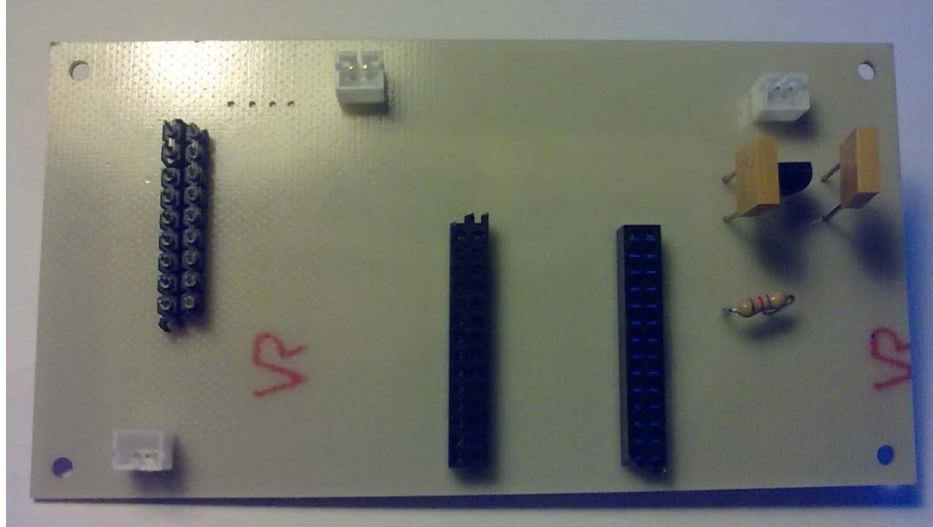
**Figura 10.2:** Esquema de implementación del Hardware

El resultado es una placa PCB de 11x7 cm. Utiliza conectores de polaridad para conectar el servo, la alimentación y la conexión serie, y tiene una entrada RJ45 para la conexión ETHERNET. Podemos ver una imagen de la placa definitiva del prototipo en la **Figura 10.3** ya conectada al servo.



**Figura 10.3:** Imagen de la placa del prototipo

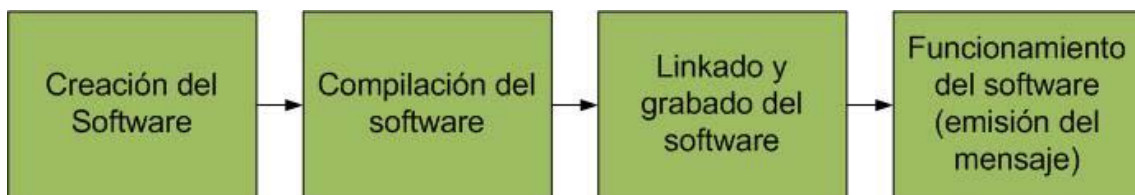
Además, está realizado utilizando pasos de forma que el módulo Ethernet y del microcontrolador se puedan quitar y poner fácilmente, y no soldar así los módulos directamente sobre la placa. Es una forma tan efectiva como soldar y que permite intercambiar los módulos si fuera necesario. La **Figura 10.4** muestra la PCB sin los módulos insertados.



**Figura 10.4:** PCB sin los módulos insertados

### 10.3 Resultados del software

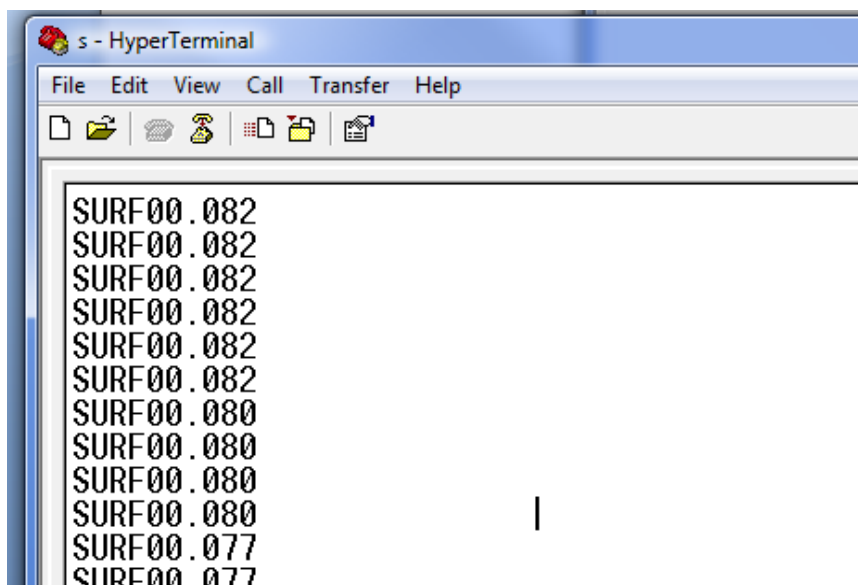
Una vez realizado el software, se compila, y se hace el linkado con el microcontrolador para grabarlo en el microcontrolador. Así, una vez recibe tensión el microcontrolador, el software comienza a funcionar y a emitir el mensaje. La **Figura 10.5** muestra un diagrama del proceso.



**Figura 10.5:** Esquema de implementación del software

Tras realizar la conexión entre el ordenador y el sistema, cuando está funcionando, un programa como HyperTerminal nos muestra datos como los de la **Figura 10.6**. En este caso, la posición que nos indica es directamente la del servo ya que no está configurado para ninguna superficie en concreto más que la del propio servo.





**Figura 10.6:** Recepción de los datos del sistema

## 10.4 Documentación

Para facilitar el desarrollo de este sistema y adaptarlo para cualquier superficie, el ANEXO 4 es un manual del sistema. En éste se detalla cómo crear desde cero el sistema, incluyendo el material necesario, la lista de archivos utilizada, como modificar las variables, etc.

Está dividido en tres grandes bloques, manual de instalación, manual de calibración y manual de usuario.

### 10.4.1 Manual de instalación

Tal y como indica su nombre, este manual explica el proceso de creación del sistema desde cero, tanto de la parte de hardware como de software. Es la parte más extensa y detallada de todas, pues es la más importante para el correcto desarrollo del sistema.

Este manual está dividido en dos secciones principales: instalación del hardware y del software.

La primera indica cómo llevar a cabo la creación del hardware a partir de los diferentes componentes. NO indica los pasos a seguir para la adaptación del sistema a la superficie, pues no es posible adaptarlo a todas las superficies posibles y detallarlo en un manual genérico. Esa parte corre a cuenta del ingeniero que esté utilizando el sistema para montarlo en la superficie deseada.

La segunda indica explica detalladamente cómo está compuesto el software y los parámetros necesarios a introducir acerca de la superficie a medir, tales como su posición mínima y máxima., identificador, etc.

#### **10.4.2      Manual de calibración**

El manual de calibración se ha desarrollado para poder aclarar la obtención de una serie de variables necesarias para la configuración e instalación del sistema. En él se explica el proceso a seguir para la obtención de las variables.

#### **10.4.3      Manual de usuario**

Este manual explica cómo obtener los datos del sistema de la forma más rápida y sencilla. Se explica la configuración del sistema y como entender los datos que vienen, poniendo ejemplos de cómo es la señal emitida y como entender los datos provenientes de ella.

## CAPÍTULO 11. CONCLUSIONES

La realización de un proyecto aeronáutico sigue una serie de procesos muy complejos que requieren de mucho tiempo de preparación, estudio y trabajo para ser llevado a cabo. En la realización de este proyecto inicialmente se buscaba realizar un sistema electrónico para poder usar en algún tipo de aplicación aeronáutica, intentando centrar la atención especialmente en la parte de electrónica. Sin embargo, el cambio de enfoque como algo más serio que una simple “práctica de electrónica” provocó tener que aprender a afrontar problemas con los que cualquier ingeniero aeronáutico deberá lidiar en su día a día y que no son únicamente de tipo técnico, sino que incluyen problemas tales como gestión del proyecto o relaciones entre diferentes departamentos y procesos.

Principalmente, el desarrollo de este proyecto nos ha aportado una nueva perspectiva de la realización de proyectos que no habíamos adquirido a lo largo de la carrera. El realizar un estudio de seguridad en paralelo con el diseño del sistema nos ha dado la capacidad de ver lo complejo que es conectar dos procesos que siguen desarrollos separados pero totalmente interrelacionados entre ellos. Además, no solo se ha comprendido la complejidad de la interconexión entre procesos sino la complejidad de cada proceso en sí, pues no se había hecho ningún trabajo igual al realizado aquí.

El resultado ha sido un trabajo con muchas horas de esfuerzo tras él, y que puede ser aprovechado para diferentes objetivos. El estudio de seguridad puede servir como ejemplo base para la realización de otros estudios o para intentar entender mejor a que se refiere con cada proceso, y su relación con el diseño. Cabe destacar que es un estudio a pequeña escala pues un estudio real de estas características no puede ser realizado en 4 meses ni por dos personas, sino por equipos de trabajo y años de trabajo. Aún así sirve como una aproximación inicial.

El diseño realizado puede ser utilizado para añadirle funcionalidades nuevas tanto al UAV del ICARUS Team como a cualquier sistema que utilice servos. No es un diseño extremadamente complejo, pero que ha conllevado muchos problemas debido precisamente a requisitos que provenían del estudio de seguridad. Se valoraron diferentes opciones y finalmente se escogió el definitivo. Nuestro sistema además está pensado para permitir ampliaciones, tal y como detalla el **ANEXO 5: Capacidad de ampliación**.

Para concluir, el desarrollo de este proyecto ha sido una buena forma de aproximarnos a lo que será trabajar en el mundo de la aeronáutica y comprobar los diversos problemas que pueden surgir, habiendo ayudado mucho el haber realizado el trabajo entre dos personas en vez de uno solo, pues en el mundo laboral lo habitual será trabajar en grupos de trabajo.

## **CAPÍTULO 12. GLOSARIO**

**FHA:** Functional Hazard Assessment

**PSSA:** Preliminary System Safety Assessment

**ZSA:** Zonal Safety Assessment

**FTA:** Fault Tree Analysis

**UAV:** Unmanned Aerial Vehicle

**UAS:** Unmanned Aerial System

**UPC:** Universitat Politècnica de Catalunya

**ADC:** Conversor analógico digital

**PC:** Ordenador Personal

**LDO:** Low Drop Out

**EMC:** Compatibilidad electromagnética

**DAL:** Design Assurance Level

**PCB:** Placa de circuito impreso

**MTBF:** Tiempo Medio Entre Fallos

## CAPÍTULO 13. BIBLIOGRAFÍA

- [1] ICARUS TEAM, *Flight Manual*, Castelldefels, 19/05/2009
- [2] "GUIDELINES AND METHODS FOR CONDUCTING THE SAFETY ASSESSMENT PROCESS ON CIVIL AIRBORNE SYSTEMS AND EQUIPMENT" (SAE ARP4761) [USA, 1996]
- [3] "CERTIFICATION CONSIDERATIONS FOR HIGHLY-INTEGRATED OR COMPLEX AIRCRAFT SYSTEMS" (SAE ARP4754) [USA, 1996]
- [4] Juan López, Pablo Royo, Cristina Barrado, Enric Pastor, *MODULAR AVIONICS FOR SEAMLESS RECONFIGURABLE UAS MISSIONS*, 30/10/08, CASTELLDEFELS
- [5] Safety Case Development Manual, EUROCONTROL
- [6] "UAS Preliminary Considerations for Classifying Hazards" (NASA-2007-TM-214539) [USA, 2007]

[END OF DOCUMENT]



**Escola Politècnica Superior  
de Castelldefels**

UNIVERSITAT POLITÈCNICA DE CATALUNYA

# **ANNEXOS**

**TÍTOL DEL TFC: Diseño y prototipado de un sistema de monitorización  
de configuración de un UAV**

**TITULACIÓ: Enginyeria Tècnica Aeronàutica, especialitat Aeronavegació**

**AUTORS: Eloy Herrera Cañadas  
Víctor de los Santos Bernad**

**DIRECTOR: Jorge Ramírez Alcántara**

**DATA: 27 de Juliol de 2009**

## ANEXO 1: Documento de planificación

	Name	Signature	Date
Proyectista 1	Eloy Herrera		12/03/09
Proyectista 2	Víctor de los Santos		12/03/09
Director	Jorge Ramírez Alcantara		12/03/09



## Registro de cambios

Fecha	Cambio
10/07/09	-Se cambia la introducción por otra que ofrece una guía más detallada de cómo elaborar un PSSA
07/04/09	- Cambios en gestión de figuras
13/06/09	-Cambio en gestión de sistemas
13/07/09	- Se cambia el formato. -Cambio en "template de documento genérico".

<b>1</b>	<b>OBJETIVO .....</b>	<b>4</b>
<b>2</b>	<b>RECURSOS HUMANOS.....</b>	<b>5</b>
<b>3</b>	<b>HERRAMIENTAS.....</b>	<b>6</b>
<b>4</b>	<b>TEMPLATE DE DOCUMENTO .....</b>	<b>7</b>
<b>5</b>	<b>TEMPLATE DE DOCUMENTO DE GESTIÓN SEMANAL.....</b>	<b>8</b>
<b>6</b>	<b>GESTIÓN DE REQUISITOS.....</b>	<b>9</b>
<b>7</b>	<b>GESTIÓN DE SISTEMAS.....</b>	<b>10</b>
<b>8</b>	<b>GESTIÓN DE TAREAS .....</b>	<b>11</b>
<b>9</b>	<b>GESTIÓN DE DOCUMENTOS.....</b>	<b>12</b>
<b>10</b>	<b>GESTIÓN DE FIGURAS.....</b>	<b>13</b>
<b>11</b>	<b>LISTA DE REQUISITOS.....</b>	<b>14</b>
<b>12</b>	<b>TAREAS E HITOS .....</b>	<b>15</b>
<b>13</b>	<b>BIBLIOGRAFÍA .....</b>	<b>18</b>
<b>14</b>	<b>GLOSARIO .....</b>	<b>19</b>

## 1 OBJETIVO

Realizar la planificación del sistema sensor que estamos diseñando.



## 2 RECURSOS HUMANOS

### **Eloy Herrera**

Estudiante de Ingeniería Técnica Aeronáutica, especialidad en Aeronavegación. Cursando el 3er y último año.

### **Víctor de los Santos**

Estudiante de Ingeniería Técnica Aeronáutica, especialidad en Aeronavegación. Cursando el 3er y último año.

Trabajó en Aertec Ingeniería y Desarrollos S.L.U. como técnico de cálculo en el período comprendido entre Enero y Marzo de 2009.

### 3 HERRAMIENTAS

- **G-Mail:** Cliente de correo electrónico proporcionado por Google. Capacidad de almacenamiento de 7GB y un máximo de 20MB en el envío de archivos adjuntos. Cada miembro implicado en la elaboración del trabajo tendrá una cuenta USER@gmail.com donde se le informará de los hitos conseguidos en el documento principal. Se puede acceder con diversos navegadores, aunque los utilizados por los miembros serán Mozilla Firefox y Google Chrome.

- **Windows Live Messenger:** Programa de comunicación por mensajes de texto instantáneo proporcionado por Microsoft. Permite la comunicación instantánea entre todos los miembros que estén agregados en la cuenta. Además permite el envío de archivos y realizar un enlace de videoconferencia entre usuarios. Se utilizará cuando se quiera comentar alguna cosa y los miembros estén conectados a la red del programa en ese momento. Los miembros deben tener todos una cuenta asociada a la red Windows Live Messenger mediante el sistema Passport de Microsoft.

- **Reuniones:** Se utilizarán para poder tratar directamente todos los asuntos relacionados con el proyecto, comentar las dudas y plantear problemas y soluciones que vayan surgiendo a lo largo del trabajo. Se realizarán con tanta asiduidad como sea necesaria, con un mínimo de una (1) reunión semanal. La reunión semanal será principalmente los Lunes a las 17:30 h., con posibilidad de cambiarlo a conveniencia de los miembros en un día puntual.

- **Google Calendar:** es una agenda y calendario electrónico desarrollado por Google. Su desarrollo todavía está en fase beta. Permite la sincronización con los contactos de Gmail de manera que podamos compartir eventos. Cada miembro implicado en el desarrollo del trabajo deberá suscribirse al calendario "TFC" donde se publicarán los plazos a cumplir para los diferentes hitos del trabajo.

- **Google Docs:** programa gratuito basado en Web para crear documentos en línea subministrado por Google. Incluye un Procesador de textos, una Hoja de cálculo y un Programa de presentación. Permite compartir los documentos con otros contactos que posean una google account de forma que estos puedan colaborar en la elaboración del documento. Cada miembro implicado en el desarrollo del trabajo deberá suscribirse al documento "TFC" y a todos aquellos que se deriven de éste.

- **Microsoft Word 2007:** Procesador de textos de Microsoft utilizado para crear documentos y darles el formato definitivo. Es un programa muy completo que permite gestionar fácilmente todo tipo de textos con figuras, referencias y tablas de forma automatizada.

- **Microsoft Excel 2007:** Es una aplicación para crear y gestionar hojas de cálculos. Nosotros lo usaremos para gestionar la lista de requisitos y para el documento de gestión semanal.

- **IAR Embedded Workbench Kickstart Edition v 4.2:** Compilador y linkador para grabar el software en código c en el microcontrolador que se utilizará en el desarrollo del sistema.

- **HyperTerminal:** Programa de comunicaciones para establecer la conexión entre el sistema desarrollado y un PC. Se utiliza como sistema de pruebas para comprobar el correcto funcionamiento. Permite realizar gran variedad de comunicaciones como conexiones de Internet, SERIE, etc.

## 4 TEMPLATE DE DOCUMENTO

### PRESENTACIÓN DE LA PÁGINA:

- Elegir margen estrecho.

### ENCABEZAMIENTO:

- La fuente para título, identificador y fecha será "Calibrí 10". Estos campos irán inscritos en la tabla que se adjunta en este documento.
- El identificador del documento estará formado por una "D" seguido de una numerología. Ejemplo: D001
- Si se trata de un Anexo el identificador estará encabezado por la letra "A" seguida del identificador del documento principal y seguida del número del anexo. Estará escrito en verde. Ejemplo: A-D001-01.
- Para referenciar un documento o anexo en posteriores documentos nos valdremos de su identificador escrito entre paréntesis.

### FIGURAS:

- Se introducirán en formato "cuadrado".
- Bajo la fotografía indicaremos el título y el identificador de la figura.
- El identificador estará encabezado por "FIG" e irá acompañado del identificador del documento y una numerología. Ejemplo: FIG-(D001-01)

### ANEXOS Y BIBLIOGRAFÍA

- Se introducirán de la forma indicada en el ejemplo.
- Se colocarán al final del documento.

Consultar template: A-D001-01.

ACTUALIZACIÓN (07/07/2009): Este template ha sido descartado a último momento. Se usará una plantilla facilitada por el cliente.

## 5 TEMPLATE DE DOCUMENTO DE GESTIÓN SEMANAL

### PRESENTACIÓN DE LA PÁGINA:

- Elegir margen estrecho.
- Orientación vertical.

### ENCABEZAMIENTO:

- La fuente para título, identificador y fecha será "Calibrí 10". Estos campos irán inscritos en la tabla que se adjunta en este documento.
- El identificador del documento estará formado por una "DS" seguido de una numerología.  
Ejemplo: DS001
- En el encabezamiento se especificará la semana a la que hace referencia el documento.

### GESTIÓN DE TAREAS

- A las tareas inacabadas se les dará el formato "Incorrecto".
- A las tareas terminadas se les dará el formato "Correcto".

Consultar template: [A-D001-02](#).

ACTUALIZACIÓN (07/03/2009): Al final se desestimó llevar a cabo una gestión del progreso semanal por el tiempo que llevaba realizarla.

## 6 GESTIÓN DE REQUISITOS

- Todos los requisitos responderán a la estructura Sujeto + Verbo + Predicado.
- Los requisitos estarán escritos en **negro**.
- Delante de cada requisito figurará su número de indicación. Éste estará entre corchetes y será un numero precedido or la letra "R". Estarán escritos en **negro**, con formato negrita.
- Debajo de cada requisito se hará una breve explicación de éste. Esta explicación estará escrita en **azul**.
- Debajo de la explicación indicaremos el identificador de la fase o tarea de la que éste se deriva. Estará escrito entre paréntesis y en **amarillo**.
- Debajo de la explicación se deberá hacer referencia a la fuente. Escribiremos la fuente con color **lila**.
- Debajo de la fuente deberemos exponer su nivel de madurez en **granate** y su estado en **verde**. Éstos parámetros irán entre corchetes.

Ejemplo:

[R001] Título del requisito  
Breve explicación del requisito  
(RW01-001)  
FUENTE: Aquí indicaremos la fuente  
[Madurez] [Estado]

- Para asegurar la trazabilidad de requisitos elaboraremos un base de datos en la que figurarán todos los requisitos.
- Cuando un requisito sea trazable a otro aparecerá a la derecha de este en la base de datos.
- La base de datos será accesible via google docs para todos los implicados en el proyecto.
- A partir de ahora en la documentación del proyecto nos referiremos a los requisitos solo por su identificador.





## 7 GESTIÓN DE SISTEMAS

- Todos los elementos que forman el sistema (subsistemas, elementos, piezas, etc.) comenzarán por **S** para saber que se trata de un elemento, subsistema, etc. y seguirán una nomenclatura de árbol similar a la del ejemplo a continuación:

Si el elemento tiene la nomenclatura S1.3.4, quiere decir que es un elemento perteneciente al subsistema 1 (S1). Los otros niveles sirven para concretar a que elemento se refiere, dentro de cada parte del subsistema. En este caso, nos referiríamos al elemento 4 de la parte 3.

- Si fuera necesario, el diagrama de árbol se extendería todo lo necesario para especificar el elemento del que se trata.
- Cada parte del sistema tiene que tener un identificador asignado.

**ACTUALIZACIÓN (13/06/2009): Esta gestión no se establecerá finalmente al no haber conseguido financiamiento para hacer los 4 sistemas y tenernos que conformar con un prototipo.**

## 8 GESTIÓN DE TAREAS

- Cada tarea principal recibe un identificador encabezado por la letra "w" y seguido por una serie numérica de dos dígitos. Ejemplo: W01.
- Estas tareas tendrán asociadas a si mismas un conjunto de subtareas cuyo identificador será el mismo que la tarea principal pero acompañado de la numerología que corresponda a dicha subtask. Ejemplo: W01-01.
- Todas las tareas estarán recojidas en el documento de planificación.
- A partir de ahora en la documentación del proyecto nos referiremos a las tareas solo por su identificador.
- Si la tarea está relacionada con un proceso del safety, el identificador de la tarea comenzará por la letra S. Ejemplo: SW01-01.
- Las verificaciones tendrán un número de tarea equivalente a la tarea que verifican, incluyendo una V delante para constatar que se trata de verificación. Ejemplo. La tarea W01-01 necesita ser verificada. La tarea de verificación tendrá el número VW01-01.

**ACTUALIZADO (07/04/2009):** Cancelado, añade mucha complejidad al proyecto. A cada tarea nueva que nos surge hay que realizar demasiados cambios en planificación y estos cambios hay que trazarlos a la resta de documentos.

## 9 GESTIÓN DE DOCUMENTOS

- A cada documento, ya sea principal o bien un anexo se le asignará un identificador.
- El identificador del documento estará formado por una "D" seguido de una numerología. Ejemplo: D001
- Si se trata de un Anexo el identificador estará encabezado por la letra "A" seguida del identificador del documento principal y seguida del número del anexo. Estará escrito en verde. Ejemplo: A-D001-01.
- Para referenciar un documento o anexo en posteriores documentos nos valdremos de su identificador escrito entre paréntesis.
- Para asegurar la correcta gestión de los documentos tendremos un documento excel donde quedarán todos recojidos.
- En este excel figurará el identificador y el título del documento.
- En este documento indicaremos a que tarea responde dicho documento.
- Cuando un documento tenga Anexos estos se indicarán en el excel debajo del documento principal y en color verde.



## 10 GESTIÓN DE FIGURAS

- A cada figura se le asignará un identificador.
- El identificador estará encabezado por "FIG" e irá acompañado del identificador del documento y una numerología. Ejemplo: FIG-(D001-01)
- Crearemos un excel donde estarán recojidas todas las figuras.
- En este excel figurará el identificador y el título de la figura.

**ACTUALIZADO (07/04/2009):** Cancelado, se llevará un registro de las figuras en cada documento independientemente del resto.



## 11 LISTA DE REQUISITOS

Consultar ANEXO 11.

## 12 TAREAS E HITOS

*NOTA: La siguiente lista de tareas fue elaborada al principio del proyecto y no se ha ido actualizando, se adjunta para ver que ha sido realizada y que los cambios al llevar a cabo un proyecto suelen hacer que el resultado diste mucho de la planificación. En caso de haber contado con más capital humano alguien se hubiera encargado del proceso de planificación y de actualizarlo a medida que el proyecto avanzaba.*

### Elaboración del documento de planificación (W01)

**Fuentes:** -SAE ARP4761.

-SAE ARP4754.

-Diapositivas aviónica.

**Productos:** -Documento de planificación (D001).

-Anexo "Template de documento" (A-D001-01).

-Diagramas Gant. (FIG-D001-ABC)

- Especificación de las herramientas a utilizar para desarrollar el proyecto (W01-01).v
- Especificación de las tareas que componen la totalidad del proyecto (W01-02).v
- Gestión de requisitos (W01-03).v
- Gestión de tareas (W01-04).v
- Especificación de requisitos (W01-05).v.
- Especificación de las pruebas que certificarán los requisitos (W01-06).v
- Especificación de los recursos humanos destinados al proyecto (W01-07).v
- Definición de un template para los documentos (W01-08).v
- Gestión de documentos (W01-09).v
- Gestión de figuras (W01-10).v
- Diagramas Gant y herramientas de planificación temporal. (W01-11).

### Elaboración del diseño arquitectural (W02)

**Fuentes:** -Especificaciones Shadow.

-.....

**Productos:** -Documento de diseño arquitectural.

- Especificaciones del sistema a integrar (W02-01).v
- Especificaciones del Shadow (W02-02).
- Estudio de las superficies de control y sus movimientos (W02-03).e
- Estudio del motor del UAV y sus parámetros (W02-04).e
- Definición del sistema sensor a implantar en las superficies de control (W02-05).
- Definición del sistema sensor a implantar en el motor (W02-06).
- Definición del sistema de conexión entre los sensores y el microcontrolador (W02-07).
- Definición del sistema de transmisión (W02-08).
- Definición del sistema que acondicionará la información recibida por los sensores (W02-09).
- Definición del sistema que mostrará la información recibida (W02-10).
- Verificación de la definición del sistema sensor a implantar en la superficie de control (VW02-05)
- Verificación de la definición del sistema sensor a implantar en el motor (VW02-06).
- Verificación de la definición del sistema de conexión entre los sensores y el microcontrolador (VW02-07).
- Verificación de la definición del sistema de transmisión (VW02-08).
- Verificación de la definición del sistema que acondicionará la información recibida por los sensores (VW02-09).
- Verificación de la definición del sistema que mostrará la información recibida (VW02-10).
- Realización del FHA (SW02-01)
- Realización Del PSSA (SW02-02)

- Realización del Particular Risk Analyses del CCA (SW02-03)
- Comienzo del Common Mode Analyses del CCA (SW02-04)

### Elaboración del diseño detallado (W03)

**Fuentes:** -Documento de diseño arquitectural

-.....

**Productos:** -Documento de diseño detallado.

- Listado de sensores que cumplen el diseño arquitectural para las superficies de control (W03-01).
- Listado de sensores que cumplen el diseño arquitectural para el motor (W03-02).
- Listado de los sistemas de conexión entre sensores y microcontrolador que cumplen el diseño arquitectural (W03-03).
- Listado de microcontroladores que cumplen el diseño arquitectural (W03-04).
- Listado de sistemas de transmisión que cumplen el diseño arquitectural (W03-05).
- Elaboración de un documento que acredite la elección de un sensor para las superficies de control (W03-06).
- Elaboración de un documento que acredite la elección de un sensor para el motor (W03-07).
- Elaboración de un documento que acredite la elección de un sistema de conexión sensor/microcontrolador (W03-08).
- Elaboración de un documento que acredite la elección de un microcontrolador (W03-09).
- Elaboración de un documento que acredite la elección de un sistema de transmisión (W03-10).
- Elaboración de un documento que acredite la elección de un sistema de recepción (W03-11).
- Descripción de la implementación del sensor de las superficies de control (W03-12).
- Descripción de la implementación del sensor del motor (W03-13).
- Descripción de la implementación del microcontrolador (W03-14).
- Descripción de la implementación del sistema de transmisión (W03-15).
- Descripción de la implementación del sistema de recepción (W03-16).
- Descripción de la integración de todos los sistemas (VW03-17).
- Verificación de la descripción de la implementación del sensor de las superficies de control (VW03-12).
- Verificación de la descripción de la implementación del sensor del motor (VW03-13).
- Verificación de la descripción de la implementación del microcontrolador (VW03-14).
- Verificación de la descripción de la implementación del sistema de transmisión (VW03-15).
- Verificación de la descripción de la implementación del sistema de recepción (VW03-16).
- Verificación de la descripción de la integración de todos los sistemas (VW03-17).
- Comienzo del SSA (SW03-01)
- Comienzo del Zonal Safety Analyses del CCA (SW03-02)
- Continuación y finalización del Common Mode Analyses del CCA(SW02-04)

### Implementación (W04)

**Fuentes:** -Documento de diseño detallado

-.....

**Productos:** -Diferentes sistemas por separado

- Especificación de las herramientas necesarias para la implementación del sistema sensor de las superficies de control (W04-01).
- Especificación de las herramientas necesarias para la implementación del sistema sensor del motor (W04-02).
- Especificación de las herramientas necesarias para la implementación del microcontrolador (W04-03).
- Especificación de las herramientas necesarias para la implementación del sistema de transmisión (W04-04).

- Obtención de los sistemas y herramientas necesarios para las diferentes implementaciones (W04-05).
- Implementación del sistema sensor para las superficies de control (W04-06).
- Implementación del sistema sensor para el motor (W04-07).
- Implementación del microcontrolador (W04-08).
- Implementación del sistema de transmisión (W04-09).
- Implementación del sistema de recepción (W04-10).
- Verificación de la implementación del sistema sensor para el motor (VW04-07).
- Verificación de la implementación del microcontrolador (VW04-08).
- Verificación de la implementación del sistema de transmisión (VW04-09).
- Verificación de la implementación del sistema de recepción (VW04-10).
- Continuación del SSA (SW03-01)
- Continuación del Zonal Safety Analyses del (SW03-02)

#### **Integración (W05)**

- Fuentes:** -Documento de diseño detallado.  
-Diferentes sistemas por separado.

**Productos:** -Sistema integrado.

- Especificación de las herramientas necesarias para la integración de los sistemas (W05-01).
- Integración de los sistemas sensores con el microcontrolador (W05-02).
- Integración del microcontrolador con el sistema de transmisión (W05-03).
- Integración de los sistemas a bordo del Shadow con la estación receptora (W05-04).
- Verificación de la integración de los sistemas sensores con el microcontrolador (VW05-02).
- Verificación de la integración del microcontrolador con el sistema de transmisión (VW05-03).
- Verificación de la integración de los sistemas a bordo del Shadow con la estación receptora (VW05-04).

#### **Defensa del proyecto (W06)**

- Fuentes:** -Toda la documentación elaborada.  
-Sistema integrado.

**Productos:** -Power Point.  
- Exposición.  
- Video.

- Recopilación de toda la documentación redactada (W06-01).
- Grabación de videos que demuestren el funcionamiento del sistema (W06-02).
- Elaboración de un power point de soporte en la exposición (W06-03).
- Preparación de la defensa (W06-04).





## 13 BIBLIOGRAFÍA

- SAE ARP4761, "Guidelines and methods for conducting the safety assessment process on civil airborne systems and equipment". U.S.A., 1996



## 14 GLOSARIO

FHA: Functional Hazard Assessment

PSSA: Preliminary System Safety Assessment

ZSA: Zonal Safety Assessment

FTA: Fault Tree Analysis

SSA: System Safety Analysis

CCA: Comon Cause Analysis

PARA: Preliminary Risk Analysis



# Diseño de Sistema Sensor Para superficies de UAV

---

Iss./Rev.: 12/07/2009

Date: 07/07/2009

[END OF DOCUMENT]

Template de documento	A-D001-01
30/03/2009	

#### PRESENTACIÓN DE LA PÁGINA:

- Elegir margen estrecho.

#### ENCABEZAMIENTO:

- La fuente para título, identificador y fecha será “Calibrí 10”. Estos campos irán inscritos en la tabla que se adjunta en este documento.
- El identificador del documento estará formado por una “D” seguido de una numerología. Ejemplo: D001
- Si se trata de un Anexo el identificador estará encabezado por la letra “A” seguida del identificador del documento principal y seguida del número del anexo. Estará escrito en **verde**. Ejemplo: **A-D001-01**.
- Para referenciar un documento o anexo en posteriores documentos nos valdremos de su identificador escrito entre paréntesis.
- 

#### FIGURAS:

- Se introducirán en formato “cuadrado”.
- Bajo la fotografía indicaremos el título y el identificador de la figura.
- El identificador estará encabezado por “FIG” e irá acompañado del identificador del documento y una numerología. Ejemplo: FIG-(D001-01).
- 

#### ANEXOS Y BIBLIOGRAFÍA

- Se introducirán de la forma indicada en el ejemplo.
- Se colocarán al final del documento.

<b>Título:</b>	<b>Identificador</b>
Fecha	

**TITULO (Calibri,14,Negrita)**

**SUBTITULO (Calibri,10,Negrita)**

Cuerpo (Calibri,10)

Enumeración:

- Ítem 1.
- Ítem 2.
- Ítem 3 (Anexo A001-01).



Ejemplo figura. [FIG-(A-D001-01)-01]

Tabla:

Abcd	Bcde	Cdef	Defg	Efgh
Ítem A1	Ítem B1	Ítem C1	Ítem D1	Ítem E1
Ítem A2	Ítem B2	Ítem C2	Ítem D2	Ítem E2
Ítem A3	Ítem B3	Ítem C3	Ítem D3	Ítem E3
Ítem A4	Ítem B4	Ítem C4	Ítem D4	Ítem E4

**ANEXOS**

Titulo del anexo. [Identificador]

**BIBLIOGRAFÍA**

Apellidos    Nombre.    “Título    del    documento”.    Edición.    Lugar    de    edición.    [Identificador]

**Tareas atrasadas:**

W01-01	Definición tarea W01-01
W01-02	Definición tarea W01-02
W01-03	Definición tarea W01-03

**Tareas de la semana:**

W02-01	Definición tarea W02-01
W02-02	Definición tarea W02-02
W02-03	Definición tarea W02-03

-Cada domingo se entregará al equipo de trabajo este documento con las tareas a realizar durante la semana siguiente.  
-Las tareas se pintarán en verde una vez completadas.

[END OF DOCUMENT]

## ANEXO 2: Diseño Arquitectural

	Name	Signature	Date
Proyectista 1	Eloy Herrera		07/07/09
Proyectista 2	Víctor de los Santos		07/07/09
Director	Jorge Ramírez Alcantara		07/07/09





Diseño de  
Sistema Sensor  
Para superficies  
de UAV

---

Iss./Rev.: D03

Date: 07/07/2009



Diseño de  
Sistema Sensor  
Para superficies  
de UAV

Iss./Rev.: D03

Date: 07/07/2009

*Registro de cambios*

Fecha	Cambio
14/05/09	-Se cambia el diagrama general del sistema - Se excluye del sistema el sistema de radioenlace propio
20/05/09	- Se elimina las funciones de monitorización del motor por haber un proyecto más completo realizándolas



## Contenido

1	INTRODUCCIÓN .....	7
2	OBJETIVO .....	8
3	ESTUDIO DEL UAV .....	9
3.1	Funcionamiento de los mecanismos de movimiento (servomotores) .....	10
3.2	Timón de profundidad .....	14
3.3	Timón de dirección .....	16
3.4	Alerón izquierdo y alerón derecho .....	17
3.5	Otras consideraciones .....	18
3.5.1	Generador de señal y rango dinámico .....	19
3.5.2	Sistema de baterías .....	20
3.5.3	Red ETHERNET MAREA .....	20
4	DISEÑO DE LA ARQUITECTURA DEL SISTEMA .....	22
4.1	S1: Posición del Timón de profundidad .....	23
4.2	S2: Posición del Timón de dirección .....	24
4.3	S3: Posición del Alerón Izquierdo .....	24
4.4	S4: Posición del Alerón Derecho .....	24
4.5	Sistema de alimentación .....	25
5	BIBLIOGRAFÍA .....	26
6	GLOSARIO .....	27



# Diseño de Sistema Sensor Para superficies de UAV

Iss./Rev.: D03

Date: 07/07/2009

## *Índice de Tablas*

TABLA 1: CARACTERÍSTICAS GENERALES DEL SHADOW .....	9
TABLA 2: RANGO DE MOVIMIENTO DEL TIMÓN DE PROFUNDIDAD .....	14
TABLA 3: RANGO DE MOVIMIENTO DEL TIMÓN DE DIRECCIÓN .....	16
TABLA 4: RANGO DE FUNCIONAMIENTO DEL ALERÓN IZQUIERDO .....	17
TABLA 5: RANGO DE FUNCIONAMIENTO DEL ALERÓN DERECHO .....	17
TABLA 6: CARACTERÍSTICAS TÉCNICAS DE LOS SERVOS DEL UAV .....	19



### *Índice de Figuras*

FIGURA 1: VISTA EN PLANTA DEL SHADOW-MK1 CON LAS SUPERFICIES RESALTADAS.....	10
FIGURA 2: SERVO MODELO JR NES-537 .....	11
FIGURA 3: ESQUEMA INTERNO DE UN SERVO .....	11
FIGURA 4: EJEMPLO DE CORRESPONDENCIA ENTRE DURACIÓN DE PULSOS Y DURACIÓN DEL SERVO.....	12
FIGURA 5: CIRCUITO RETROALIMENTADO (LAZO CERRADO) DE FUNCIONAMIENTO DEL SERVO .....	13
FIGURA 6: TRANSMISIÓN DEL MOVIMIENTO DEL SERVO A LA SUPERFICIE .....	14
FIGURA 7: FOTO DE LA CONEXIÓN ENTRE LOS SERVOS Y LA SUPERFICIE DE CONTROL MEDIANTE VARILLAS .....	15
FIGURA 8: PLANTA DE LA CAVIDAD DEL SERVO DEL TIMÓN DE PROFUNDIDAD .....	15
FIGURA 9: ALZADO DE LA CAVIDAD DEL SERVO DEL TIMÓN DE PROFUNDIDAD.....	16
FIGURA 10: FOTOGRAFÍA DE LA CONEXIÓN DEL SERVO CON LA SUPERFICIE DE CONTROL.....	17
FIGURA 11: DETALLE DE LA CONEXIÓN ENTRE EL SERVO Y LA VARILLA QUE VA A LA SUPERFICIE DE VUELO .....	18
FIGURA 12: PLANTA DE LA CAVIDAD DEL SERVO DEL ALERÓN .....	18
FIGURA 13: ALZADO DE LA CAVIDAD DEL SERVO DEL ALERÓN.....	19
FIGURA 14: ESQUEMA DE LA RED MAREA.....	21
FIGURA 15: DISEÑO ARQUITECTURAL DEL SISTEMA .....	22
FIGURA 16: DIAGRAMAS DE SUBSISTEMAS.....	22
FIGURA 17: DISTRIBUCIÓN DE LA ALIMENTACIÓN .....	25



## 1 INTRODUCCIÓN

El cliente ha pedido un sistema capaz de leer la posición de las superficies de vuelo de un UAV y transmitirlos para ser monitorizadas. Para ello se debe realizar un estudio detallado del funcionamiento de las superficies de vuelo que se quieren monitorizar, estudiar las distintas posibilidades y desarrollar un diseño de ello. Además, hay que estudiar la zona donde debe ir instalado el sistema para conocer las limitaciones y las medidas más oportunas para llevar a cabo el sistema. El diseño estará totalmente sujeto a los requisitos de SAFETY que se han desarrollado y por tanto se tendrán totalmente en cuenta.



## 2 OBJETIVO

Realizar el diseño de un sistema capaz de medir la posición de las superficies de vuelo de un UAV y transmitirla para ser monitorizada.

### 3 ESTUDIO DEL UAV

La realización del sistema va totalmente ligada al sistema al que se le va a añadir. Es muy importante conocer las especificaciones para poder realizar correctamente el sistema. En este apartado se realizará un estudio general del UAV y de las zonas donde debe ir implementado el sistema.

En la Tabla 1 se pueden ver las características generales del UAV SHADOW MK-1 [1], tales como tiempo máximo de vuelo, la tensión de salida de las baterías para instrumentación del UAV, etc. En el futuro, se espera implementar una red ETHERNET (proyecto MAREA) para todo el sistema.

Modelo	SHADOW MK-1
Envergadura Alar	5,050 m
Longitud	3,594 m
Altura	0,89 m
OEW	55 kg
MTOW	90 kg
Carga de Pago (Payload)	Entre 15 y 25 kg
Potencia	22 HP
Rango de velocidad	Entre 75 y 208 km/h
Alcance máximo	200 km
Tiempo máximo de operación	6 horas
Baterías para instrumentación	5 V

Tabla 1: Características generales del SHADOW

Según los requisitos R001, R002, R003 y R004 [2], las superficies que deben controlar nuestro sistema serán el timón de profundidad, timón de dirección y alerón izquierdo y derecho. Esas superficies, por tanto, deben ser estudiadas para conocer como se debe realizar el sistema. En la Figura 1 se puede ver una vista en planta del SHADOW-MK1 con la posición de cada superficie a controlar remarcada con un círculo y el mecanismo que mueve a la superficie localizado con una cruz (cada superficie y su respectivo accionador están resaltados del mismo color y siendo diferentes entre superficies).



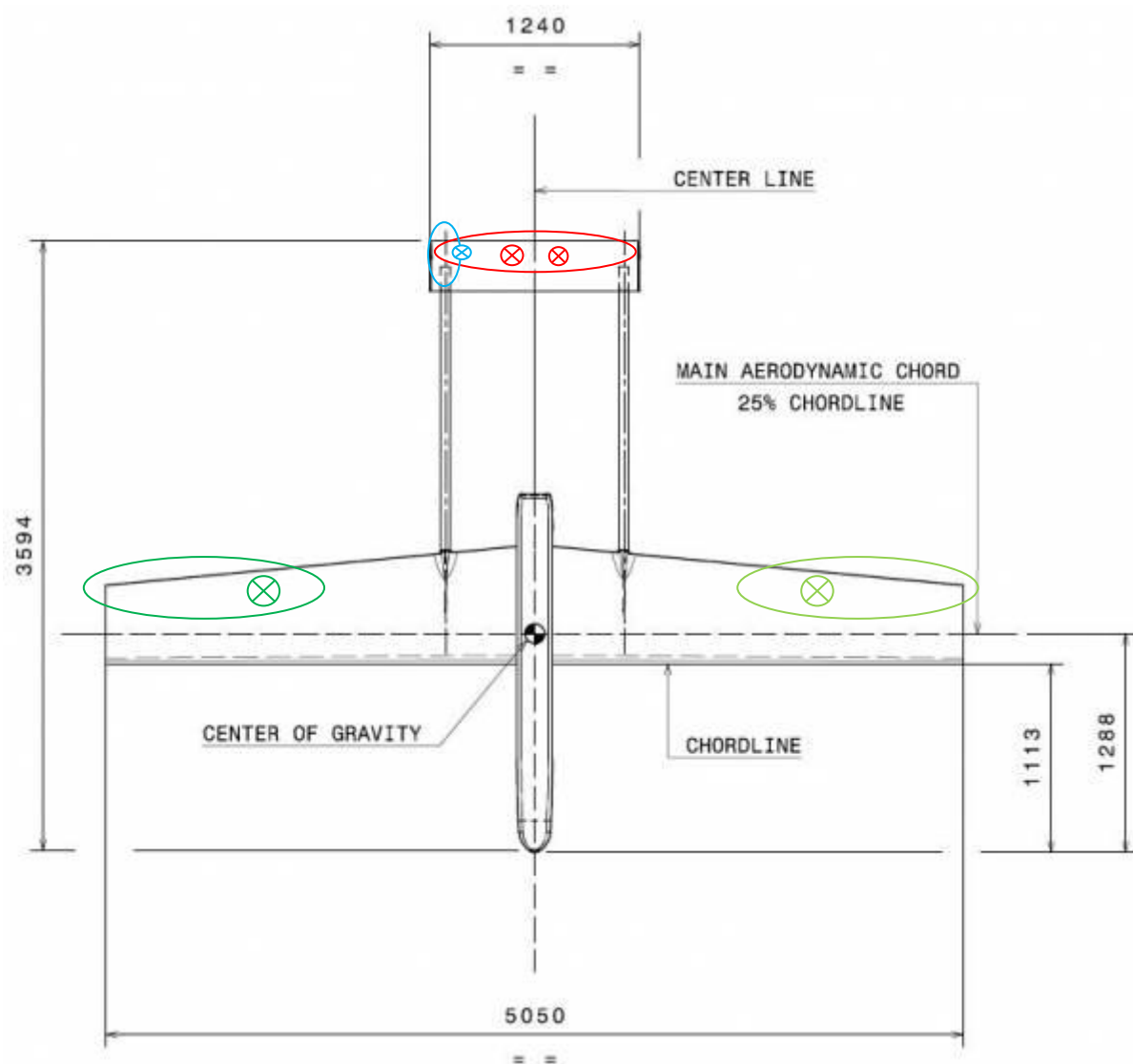


Figura 1: Vista en planta del SHADOW-MK1 con las superficies resaltadas

### 3.1 Funcionamiento de los mecanismos de movimiento (servomotores)

El estudio del sistema nos ha llevado a ver que todas las superficies de control están accionadas mediante servomotores. Los servomotores (a partir de ahora solo servos) son actuadores que se colocan en la posición que nosotros deseamos y son capaces de mantenerse en esa posición. Los servos trabajan dentro de un rango de movimiento limitado debido a su diseño, aunque con modificaciones ese rango es capaz de cambiar (modificaciones no necesarias para el proyecto a desarrollar). Suelen tener un rango de funcionamiento de  $180^\circ$  (medio giro), aunque la mayoría permiten un rango ligeramente superior de hasta  $210^\circ$ . En la Figura 2 vemos una fotografía de un servo estándar como los instalados en el UAV (éste en concreto, como se verá en siguientes apartados, es el servo que acciona el timón de profundidad).



Figura 2: Servo modelo JR NES-537

Los servos están compuestos por un motor de corriente continua, un circuito de control y una caja reductora. El motor el encargado de realizar el giro al aplicarse una tensión entre los dos terminales de éste. Son motores de alta velocidad, pero producen un par muy pequeño (el par de un motor es la "fuerza" con la que gira el motor). Para solucionar este problema, se incorpora a los servos una caja reductora encargada de aumentar ese par mediante la transformación de gran parte de esa velocidad de giro del motor en torsión. Finalmente, se encuentra el circuito de control, que es el encargado de controlar el giro del motor y comprobar que está en la posición correcta. El circuito de control está formado por una placa controladora y un potenciómetro. En la Figura 3 vemos un dibujo del esquema interno de un servo [2].

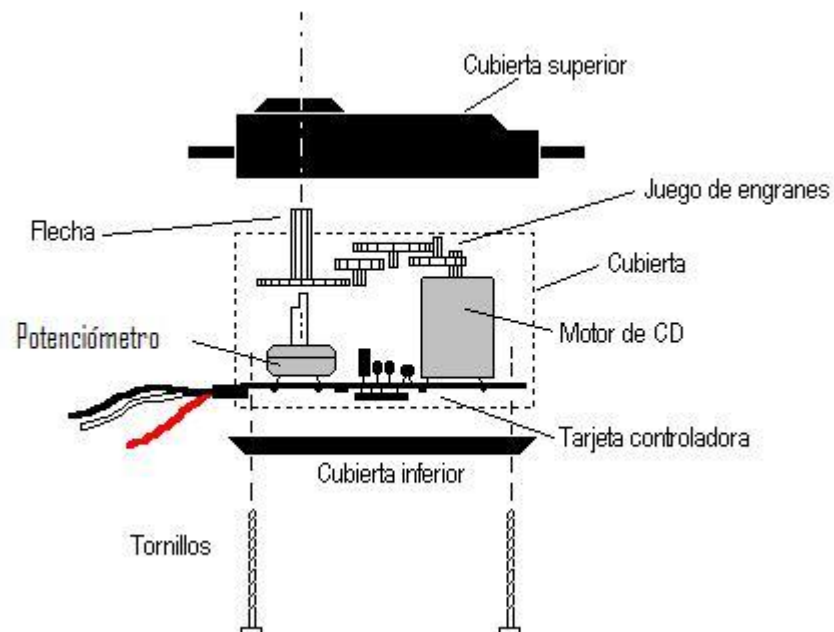


Figura 3: Esquema interno de un servo

El servo funciona mediante la recepción de pulsos. La mayoría de servos funcionan mediante señales de control denominadas *Pulse Width Modulated* (PWM a partir de ahora). La idea de funcionamiento se basa en la recepción de pulsos que indican la posición que debe obtener el servo en función de la duración del pulso. Utilizaremos el servo del UAV para ilustrar un ejemplo en el que se entienda mejor el funcionamiento de este servo.

Según las especificaciones del servo JR NES-537, tiene un movimiento comprendido en el rango de 0° a 180°, correspondiendo una señal de pulso de 1 y 2 ms (milisegundos) respectivamente. Esto quiere decir, que si recibe un pulso de duración 1 ms. el servo se colocará en la posición de 0°, y si recibe una señal de 2 ms. en 180°. Se puede realizar una ecuación lineal para conocer fácilmente a qué posición corresponde cada duración del pulso:

$$(y-y_0)/(x-x_0)=(y_1-y_0)/(x_1-x_0);$$

$$(y-0^\circ)/(x-1\text{ ms})=(180^\circ-0^\circ)/(2\text{ ms}-1\text{ ms});$$

$$y=180^\circ \cdot x - 180^\circ;$$

Siendo  $y$  la posición en grados del servo y  $x$  la duración del pulso en milisegundos. De esta forma, sabemos que con un pulso de 1.5 ms. tenemos al servo en una posición de 90°. Y así cualquier posición. En la Figura 4 vemos un ejemplo con la señal en tiempo y la posición del servo en función de esta.

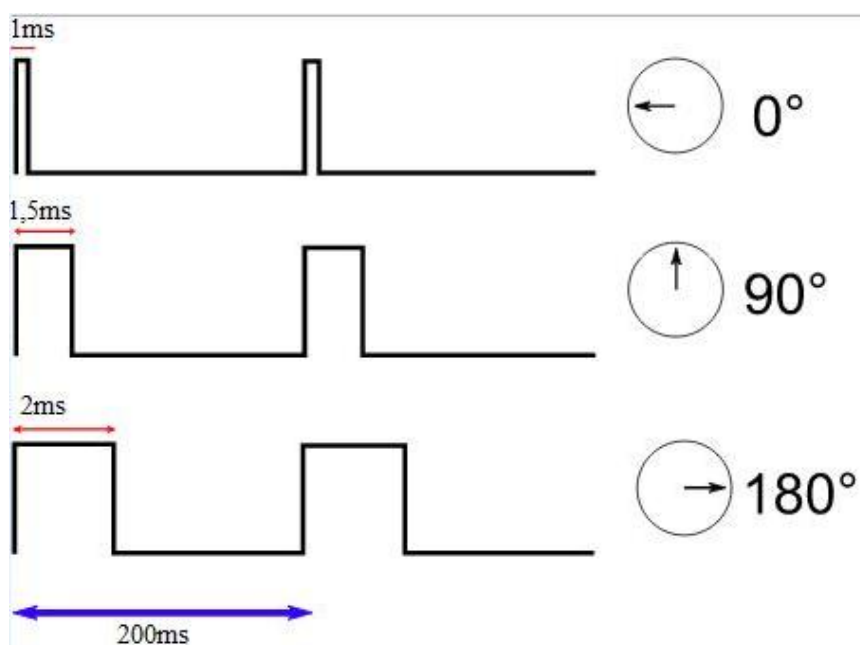


Figura 4: Ejemplo de correspondencia entre duración de pulsos y duración del servo.

El servo recibe un pulso de una duración determinada. El circuito de control del servo entonces es el encargado de mover el motor hasta la posición deseada. Hay que aclarar que el movimiento del servo no viene directamente del eje del motor, sino que antes pasa por la caja reductora para añadirle la potencia, tal y como se ve en la figura 1 añadida anteriormente. Para controlar que el servo se ha colocado en la posición deseada, se utiliza un circuito de lazo cerrado. Este circuito funciona mediante el uso de un sensor resistivo, un potenciómetro. El potenciómetro es un tipo de sensor resistivo con el que, gracias a un movimiento mecánico, la resistencia que tiene va variando. Hay varios tipos de potenciómetros, pero el potenciómetro de los servos es un potenciómetro que mide el desplazamiento angular. Además, es un potenciómetro lineal, ya que la posición angular del potenciómetro es directamente proporcional a la resistencia de este. El circuito de control conoce qué valor tiene que tener la resistencia del potenciómetro para cada ángulo (en verdad, el circuito de control controla la tensión que hay en el potenciómetro). El sensor se encuentra en el eje del servo, es decir, está situado en el eje del que obtenemos el movimiento deseado.

El circuito de control sabe en qué posición tiene que estar el servo a cada momento, y controla que el servo esté en esa posición mediante el potenciómetro. Lo que hace es controlar que la resistencia que tiene el potenciómetro concuerda con el ángulo que tiene que tener el servo. Si no es así, el circuito de control se encarga de mover el motor hasta que el potenciómetro está en la posición deseada, ya que eso quiere decir que el servo está en esa posición deseada. En la Figura 5 se puede ver un diagrama del circuito retroalimentado del servo.

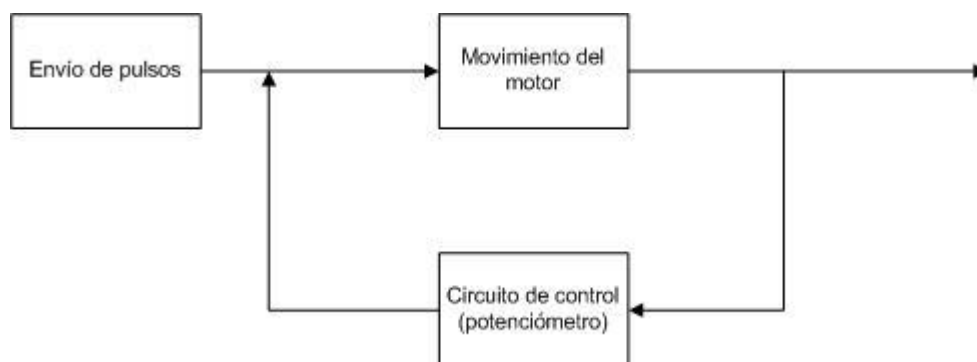


Figura 5: Circuito retroalimentado (lazo cerrado) de funcionamiento del servo





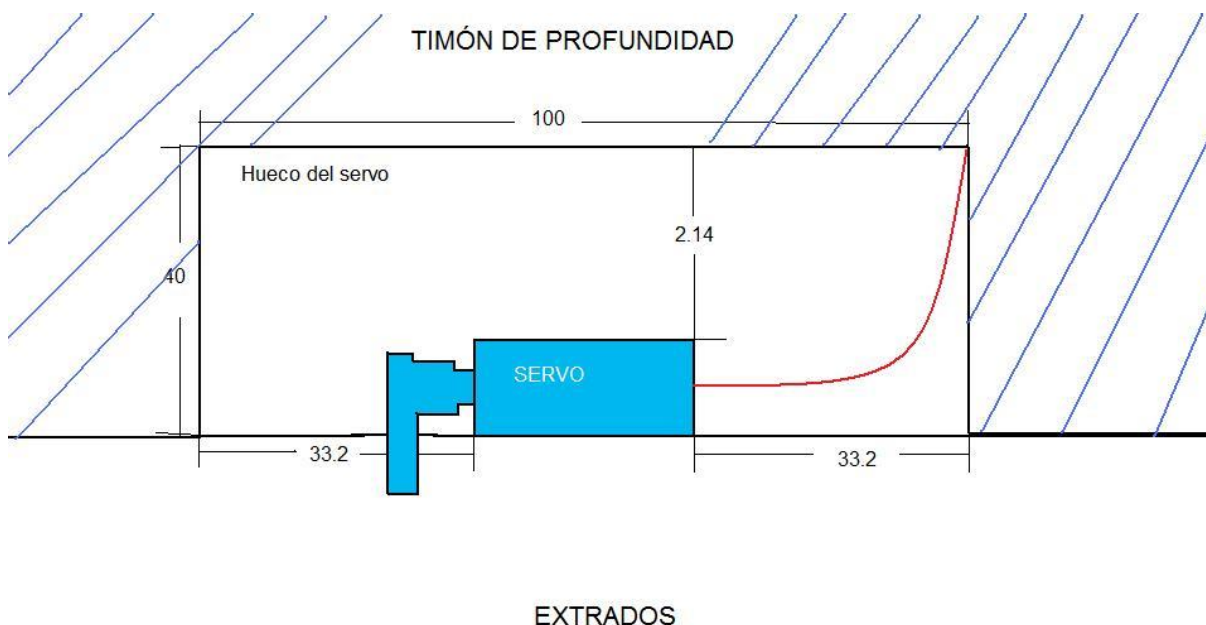


Figura 9: Alzado de la cavidad del servo del timón de profundidad

### 3.3 Timón de dirección

El timón de dirección es el encargado del movimiento de *yaw* del avión. En caso de fallo de los alerones, sería posible controlar la dirección del UAV gracias a este sistema. Se utiliza sobre todo para estabilizar el avión y controlar su posición.

El rango de movimiento del timón de dirección se puede ver en la Tabla 3. Es necesario indicar que la superficie permite el movimiento de su posición mínima y máxima hasta un 20% de su posición especificada (5° por debajo y por encima respectivamente).

Máximo	-15°
Neutro	0
Mínimo	-15°

Tabla 3: Rango de movimiento del timón de dirección

La Figura 10 es una fotografía de la conexión entre el servo y el timón de profundidad donde se puede apreciar fácilmente como está realizada la conexión.

Figura 10: Fotografía de la conexión del servo con la superficie de control

Existe un problema con la localización de este servo y es que se encuentra en un hueco totalmente adaptado al tamaño de esto. Por tanto, para implementar el sistema en esta superficie sería necesario realizar una modificación en el UAV, aumentando el hueco en el que se encuentra o colocando el superficie en una zona con más espacio y que se conecte mediante un cable desde el sistema hasta la localización del sensor que medirá la posición

### 3.4 Alerón izquierdo y alerón derecho

Debido a la casi simetría entre el alerón derecho y el izquierdo se ha decidido agruparlos en un mismo apartado. Los alerones controlan el movimiento de *roll*



Los alerones funcionan con sendos servos HS-700B [1]. En la Figura 11 se puede apreciar un de la conexión entre el servo y la varilla que va al alerón.



Figura 11: Detalle de la conexión entre el servo y la varilla que va a la superficie de vuelo

En la Figura 8 y la Figura 9 podemos ver un esquema de la planta y el alzado del hueco donde va alojado el servo del alerón (sigue el mismo esquema en ambos alerones).

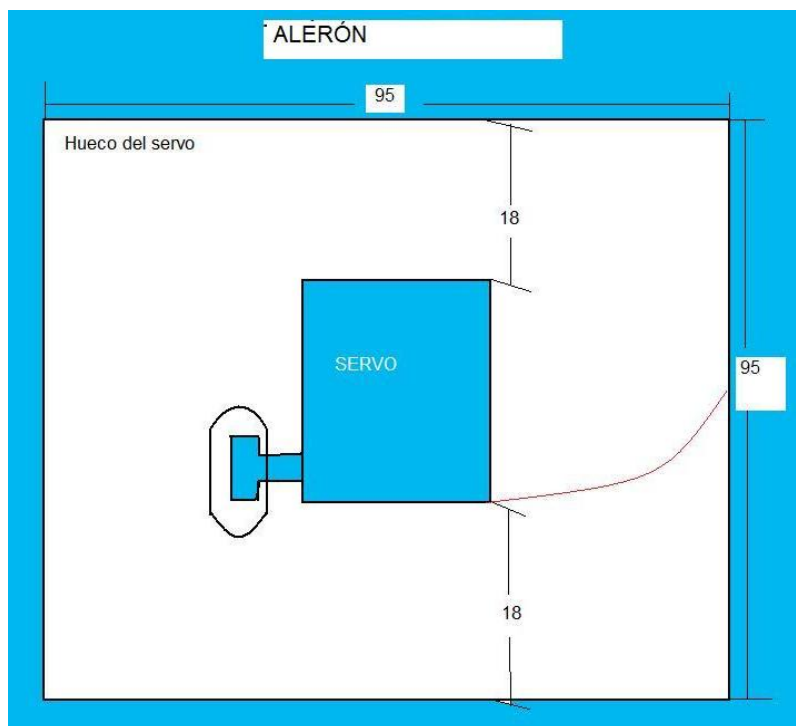


Figura 12: Planta de la cavidad del servo del alerón

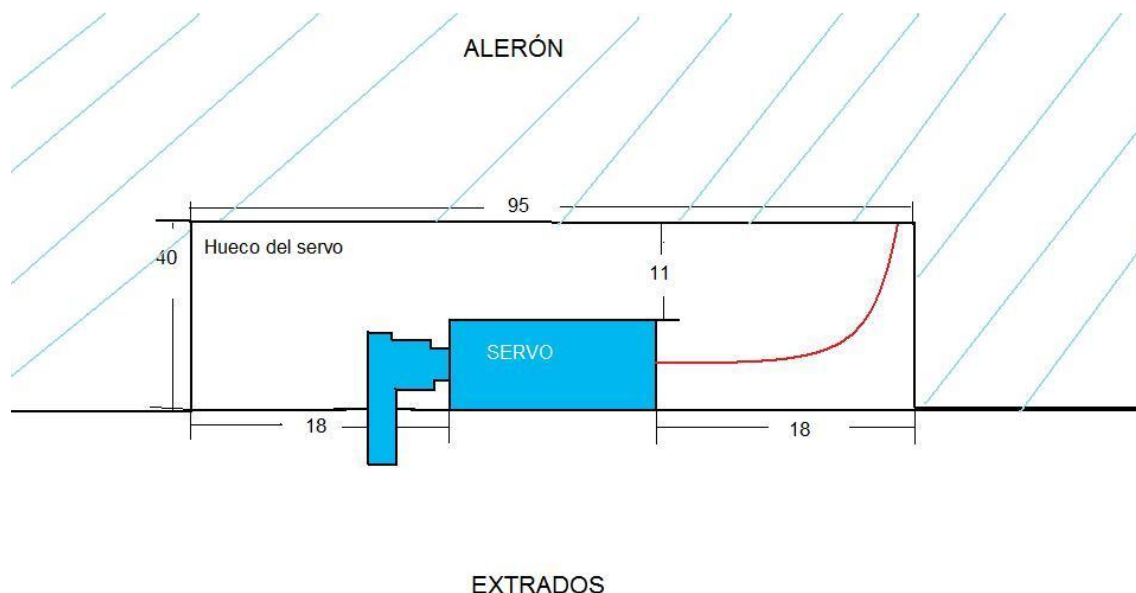


Figura 13: Alzado de la cavidad del servo del alerón

### 3.5 Otras consideraciones

#### 3.5.1 Generador de señal y rango dinámico

El sistema de servos funciona mediante la recepción de pulsos tal y como se ha explicado en el apartado del funcionamiento de los servos. Cada marca de cada servo tiene un sistema propio y que convierte la duración de cada pulso en una posición para el servo. De los servos utilizados, la Tabla 6 indica la posición mínima, neutral y máxima de los servos en función de la duración del pulso recibido, y la frecuencia de funcionamiento [2].

Fabricante	Mínima (0°)	Neutral (90°)	Máxima (180°)	Frecuencia
Hitech (HS-700B)	0,9 ms	1,5 ms	2,1 ms	50 Hz
JR	0,8 ms	1,5 ms	2,2 ms	50 Hz

Tabla 6: Características técnicas de los servos del UAV

Hay que tener en cuenta no solo los receptores de los pulsos, los servos, sino también el generador de pulsos. Según los datos del manual del UAV [1], el generador de señales tiene una resolución de 50  $\mu$ S. A partir de ahí se puede encontrar el Rango dinámico para poder obtener fácilmente la resolución de cualquier parte del sistema. Para ello, utilizamos el peor caso, es decir, el que tenga el rango dinámico más bajo, pues si cogiéramos otro el error que obtendríamos sería mayor a la resolución del sistema.

El rango dinámico se calcula mediante la siguiente ecuación:

$$R_d = 20 \log \frac{4 \cdot J_{CK}}{4 \cdot A_{CK} \cdot 0.5 \cdot J}$$

Así, y utilizando los datos del servo HS-700B (marca Hitec) obtenemos el rango dinámico:

$$R_d = 20 \log \frac{2100 \cdot 900}{50} = 27.6 \text{ dB}$$

### 3.5.2 Sistema de baterías

El UAV ya contiene un sistema de baterías que se puede utilizar para la instrumentación. Estas baterías, de 5 V pueden ser utilizadas para alimentar nuestro sistema interno.

### 3.5.3 Red ETHERNET MAREA

La arquitectura de este sistema se conoce como *Integrated Modular Avionics* (IMA). Es una arquitectura que se utiliza mucho en los sistemas aviónicos actuales, en las que las diferentes partes aviónicas comparten un único entorno de computación. Así, la aviónica interna no solo controlará el vuelo y la navegación del UAV, sino la carga de pago y la misión de éste. En el caso que nos ocupa, el sistema funciona mediante la división de elementos distribuidos, conocidos como servicios, interconectados mediante un sistema de comunicaciones interno, conocido como MAREA. Los diferentes servicios se colocan sobre los diferentes nodos computacionales mediante una red de bajo coste como es ETHERNET. Eso da una gran flexibilidad al sistema.

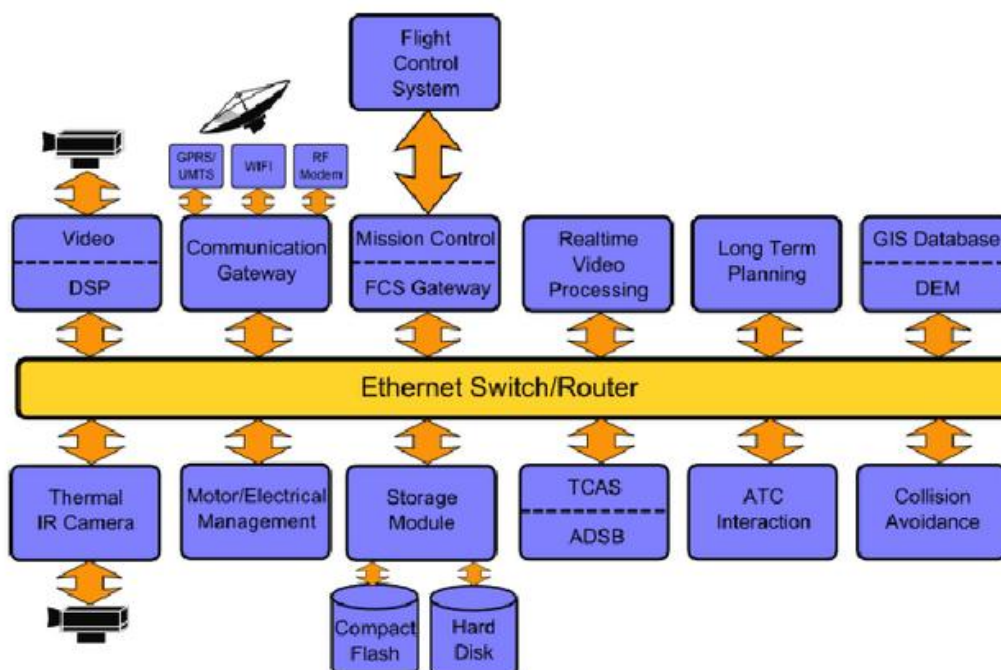


Figura 14: Esquema de la red MAREA

La Figura 14 muestra la arquitectura basada en la interconexión de nodos a través de una red Ethernet. Actualmente, esta arquitectura está solo implementada a nivel computacional, pero se espera más adelante adaptarla a un sistema de microcontroladores para poder ser embarcada en el UAV. [3].

## 4 DISEÑO DE LA ARQUITECTURA DEL SISTEMA

La realización del sistema sensor es muy parecida a cualquier otro sistema de monitorización de parámetros. La Figura 15 muestra el esquema típico de un diseño arquitectural de un sistema de sensores. En este caso, tras el estudio del UAV se decide que la mejor opción es medir la posición del servo y sacar a partir de ella la posición de la superficie.

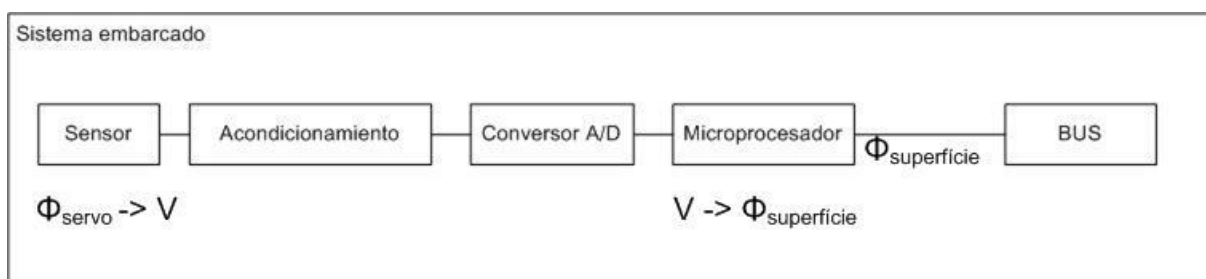


Figura 15: Diseño Arquitectural del sistema

Se decide la realización de un sistema genérico capaz de ser utilizado en cualquier superficie de vuelo controlada por un servo. De esta forma, se puede realizar un único sistema que, preparándolo adecuadamente a cada superficie, se puede adaptar fácilmente a cada una de estas.

El sistema está compuesto por 4 subsistemas principales tal y como representa la Figura 16.

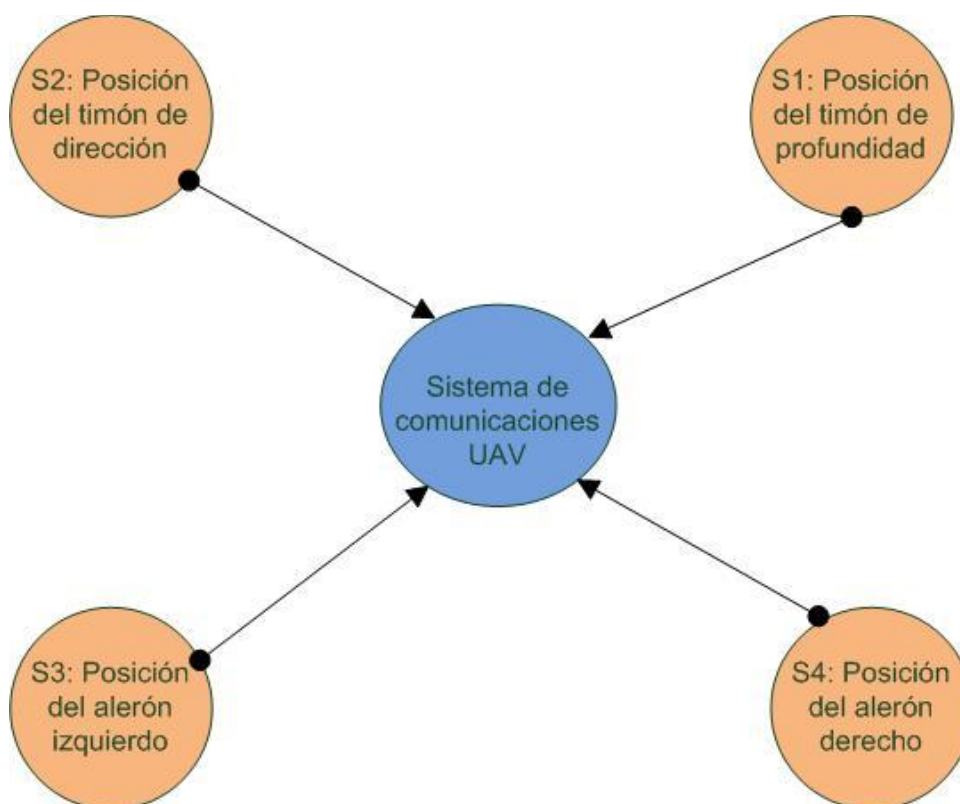


Figura 16: Diagramas de subsistemas

Los subsistemas S1, S2, S3 y S4 se localizarán en la zona del servo que controla la superficie para asegurar que los datos recibidos del sistema sean correctos y no haya problemas con la recepción de los datos (se asegura la proximidad entre el servo y el sistema).

De los subsistemas S1, S2, S3 y S4 derivan los requisitos R009, R010, R011 y R012 respectivamente. [3]. Estos requisitos designan la función que debe realizar cada subsistema. Mediante la realización de un sistema sensor como el detallado al principio de este apartado se consigue cumplir esos requisitos. Además, todo el sistema general debe cumplir los requisitos R006, R007 y R008, ya que son aplicables para todo el sistema indistintamente del subsistema.

R006: Este requisito trata sobre el peso máximo del avión. El peso de nuestro sistema debe ser extraído de la carga de pago ya que el peso del UAV no contempla el peso de nuestro sistema. Por tanto, nuestro sistema no puede exceder los 25 kg máximos de la carga de pago [1].

R007: El sistema debe caber en las zonas donde se encuentran los sistemas de los que se quieren obtener los datos. Se realizará un diseño en función del espacio disponible en cada superficie para la realización de nuestro sistema.

R008: Independencia entre subsistemas. La utilización de un único sistema genérico aplicable a cada subsistema lo hace independiente de uno y otro, y por tanto, este requisito se cumple desde el momento en que se ha desarrollado el esquema del sistema.

R014: Este requisito especifica, debido al futuro funcionamiento del UAV mediante una red de Ethernet, que el sistema debe ser capaz de transmitir los datos por la red de Ethernet. Por tanto cada subsistema debe ser capaz de enviar datos a través de la red de ETHERNET y tener su propia dirección física, IP, etc. Además, el requisito R017 establece una conexión SERIE para realizar pruebas de conexión.

R015: Durante el estudio del UAV se ve que el rango dinámico del sistema de control del UAV es de 27.6 dB. Este debe ser el rango dinámico de cada subsistema.

#### 4.1 S1: Posición del Timón de profundidad

El subsistema S1, encargado de monitorizar el timón de profundidad tiene un rango de funcionamiento entre 0° y 25°, con capacidad de aumentar en hasta 10° su rango. Por tanto, tendrá un rango máximo de 35°, de la cual, cumpliendo el requisito R015, obtenemos la resolución del sistema:

$$\Delta \theta = \frac{35^\circ}{24} = 1.46^\circ$$

Así, la resolución del sistema deberá ser de 1.46°.









## 5 BIBLIOGRAFÍA

- [1] ICARUS TEAM, *SHADOW MK-1 FLIGHT MANUAL*. 19/05/2009, CASTELLDEFELS
- [2] *Servomotores*, Yamid Ramirez, <http://www.monografias.com/trabajos60/servo-motores/servo-motores.shtml>, última fecha de revisión: 11/07/09
- [3] Juan López, Pablo Royo, Cristina Barrado, Enric Pastor, *MODULAR AVIONICS FOR SEAMLESS RECONFIGURABLE UAS MISSIONS*, 30/10/08, CASTELLDEFELS



## 6 GLOSARIO

**UAV:** Unmanned Aerial Vehicle (Vehículo aéreo NO tripulado)

**OEW:** Operative Empty Weight (Peso operative en vacío)

**MTOW:** Maximum Take Off Weight (Peso Máximo de Despegue)

**HP:** Horse Power (Caballos de potencia)

**DR:** Dynamic Range (Rango dinámico)

**IMA:** Integrated Modular Avionics (Aviónica modular integrada)

**PWM:** Pulsed Width Modulated (Modulación por ancho de pulsos)



Diseño de  
Sistema Sensor  
Para superficies  
de UAV

Iss./Rev.: D03

Date: 07/07/2009

---

[END OF DOCUMENT]

## ANEXO 3: Diseño detallado del sistema

	Name	Signature	Date
Proyectista 1	Eloy Herrera		20/06/09
Proyectista 2	Víctor de los Santos		20/06/09
Director	Jorge Ramírez Alcantara		20/06/09



## Diseño de Sistema Sensor Para superficies de UAV

Ref.:  
Iss./Rev.: D03  
Date: 20/06/2009

### *Registro de cambios*

Fecha	Cambio
14/05/09	-Se decide cambiar el diseño por uno que utilice sensores capacitivos
01/06/09	- Se cambia el diseño por uno que utilice el sensor interno del servo



# Diseño de Sistema Sensor Para superficies de UAV

Ref.:  
Iss./Rev.: D03  
Date: 20/06/2009

<i>Contenido</i>	
1	Introducción..... 6
2	Objetivo DEL PROYECTO ..... 7
3	DISEÑO DEL SISTEMA ..... 8
3.1	Funcionamiento de las superficies de control..... 8
3.2	Principio De Funcionamiento Del Sistema Sensor ..... 11
3.3	Elección Del microcontrolador ..... 12
3.4	Conexión de nuestro sistema sensor al actuador ..... 13
3.5	Conexión del potenciómetro al microcontrolador ..... 15
3.6	Transmisión de datos..... 15
3.7	Alimentación y consumo ..... 16
4	DISEÑO DE SOFTWARE ..... 18
4.1	Función 1: Recepción de los datos del sensor [R030] ..... 19
4.2	Función 2: Amplificación de la señal recibida [R031]..... 19
4.3	Función 3: Conversión A/D [R032] ..... 19
4.4	Función 4: Tratamiento de los datos [R033]..... 20
4.5	Función 5: Transmisión de los datos en serie [R034] ..... 22
4.6	Función 6: Transmisión de los datos por ETHERNET [R035]..... 23
4.7	Distribución de los ficheros que forman el software ..... 24
5	Conexiones del sistema ..... 26
6	Relación del diseño con los aspectos de SAFETY..... 28
6.1	Requisitos del Preliminary System Safety Assessment..... 28
6.2	Requisitos del Zonal Safety Assessment..... 28
7	Referencias ..... 31
8	GLOSARIO ..... 32



# Diseño de Sistema Sensor Para superficies de UAV

Ref.:  
Iss./Rev.: D03  
Date: 20/06/2009

## Índice de Tablas

Tabla 1: Consumo del Sistema.....	16
Tabla 2: Propiedades de la conexión SERIE .....	22

### *Índice de figuras*

Figura 1: Esquema interno de un Servo .....	8
Figura 2: Distribución de Servos en el UAV [2] .....	9
Figura 3: Transmisión del servo a la superficie de vuelo [2].....	9
Figura 4: Interior de un servo con el potenciómetro remarcado .....	10
Figura 5: Diagrama de bloques del Sistema Sensor .....	11
Figura 6: Detalle de la conexión del Potenciómetro al circuito.....	13
Figura 7: Zona a perforar para sacar los cables que van al potenciómetro .....	13
Figura 8: Esquema del servo con la modificación .....	14
Figura 9: Servo con los cables soldados y cerrado .....	14
Figura 10: Diagrama de conexiones USCI del microcontrolador .....	15
Figura 11: Acondicionamiento del regulador MCP1702 .....	16
Figura 12: Esquema de componentes del sistema sensor .....	17
Figura 13: Diagrama de funciones.....	18
Figura 14: Algoritmo de inicialización de la variable posmin y posmax.....	20
Figura 15: Esquema de la red y relación con nuestro sistema .....	23
Figura 16: Distribución de ficheros de software .....	24
Figura 17: Esquema de conexiones del sistema .....	26
Figura 18: Esquema de la PCB utilizada para el prototipo .....	27
Figura 19: PCB ensamblada .....	27
Figura 20: Funcionalidades críticas de la aeronave.....	29
Figura 21: Esquema de un amplificador.....	29





## 1 INTRODUCCIÓN

El avión no tripulado conocido como Shadow, no posee ningún sistema capaz de informar de la posición que tiene cada superficie de vuelo a cada momento. Debido a eso, se decide crear un sistema de sensores que sea capaz de obtener esa información.

La utilización de este sistema de sensores incrementará la seguridad del UAV gracias a que permitirá conocer mejor el estado de las superficies de vuelo, encargadas del movimiento del avión.

Se decide utilizar un sistema genérico y no uno concreto para cada superficie debido a la similitud del funcionamiento de todas las superficies, pudiendo aprovechar un único sistema fácilmente reprogramable por Software tal y como se indicará en un manual del sistema, donde lo único que habrá que cambiar serán los parámetros de desplazamiento de cada superficie de vuelo.

Este documento detallará el diseño del prototipo que se va a utilizar.



## Diseño de Sistema Sensor Para superficies de UAV

Ref.:  
Iss./Rev.: D03  
Date: 20/06/2009

## 2 OBJETIVO DEL PROYECTO

Diseño detallado del sistema que se va a utilizar, tanto de la parte de hardware como del software.

### 3 DISEÑO DEL SISTEMA

#### 3.1 Funcionamiento de las superficies de control

Las superficies de control funcionan mediante la utilización de servomotores (o simplemente servos como los llamaremos a partir de ahora). Los servos son sistemas que permiten mover superficies de forma precisa en función de las señales que reciben. En la Figura 1 se puede ver el esquema de funcionamiento de un servo [1]

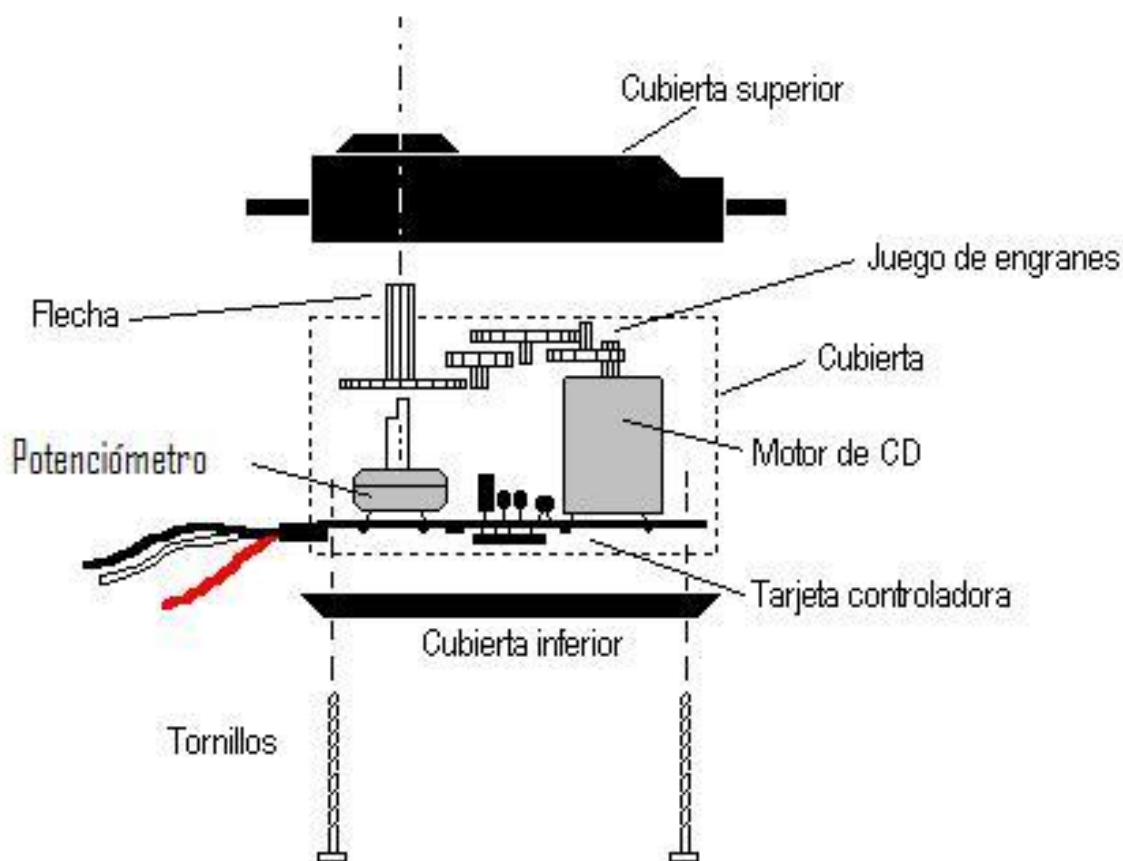


Figura 1: Esquema interno de un Servo

En el UAV, los servos son estándar, y funcionan exactamente de la misma forma. Están distribuidos por el UAV para mover las superficies. Hay un servo en el alerón derecho, otro en el alerón izquierdo, uno encargado del timón de dirección y dos dedicados al timón de profundidad. La distribución en el UAV se puede ver en la Figura 2.

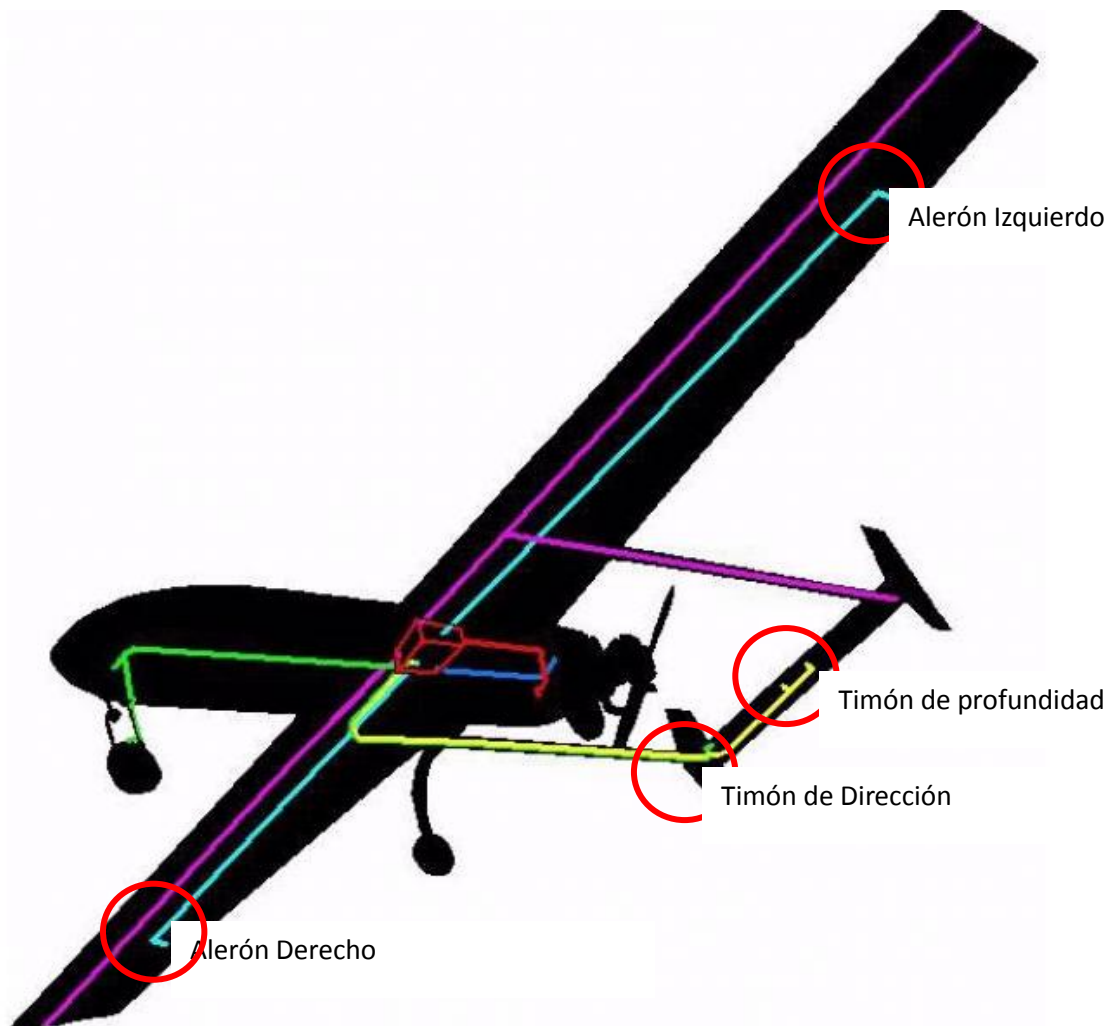


Figura 2: Distribución de Servos en el UAV [2]

El servo tiene un sistema interno que manda al motor moverse en función de la señal recibida. El motor gira, moviendo una serie de ruedas dentadas para ganar un momento mayor, hasta finalmente girar la rueda que gira el eje del servo. Este movimiento angular produce el movimiento de la superficie gracias a unas varillas unidas entre la superficie y el servo, como ejemplifica la Figura 3.

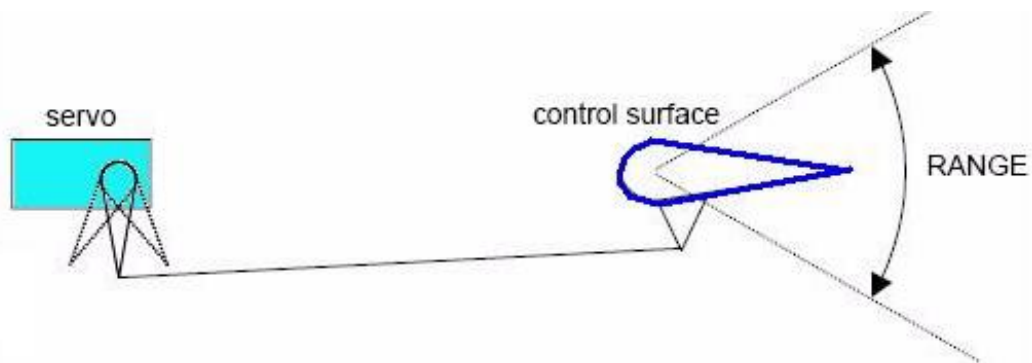


Figura 3: Transmisión del servo a la superficie de vuelo [2]

El servo, además, controla la posición en la que está a cada momento el servo y ajustándose a lo requerido. Para ello, utiliza un circuito de lazo cerrado, con un sensor interno angular, un potenciómetro, que indica a cada momento la posición que tiene el servo. Este potenciómetro está directamente alineado con el eje del servo y gira de forma solidaria con este. El potenciómetro está conectado a una placa, que se encarga de procesar la información recibida por éste, y de ahí sabe la posición. Se puede ver en la Figura 4.

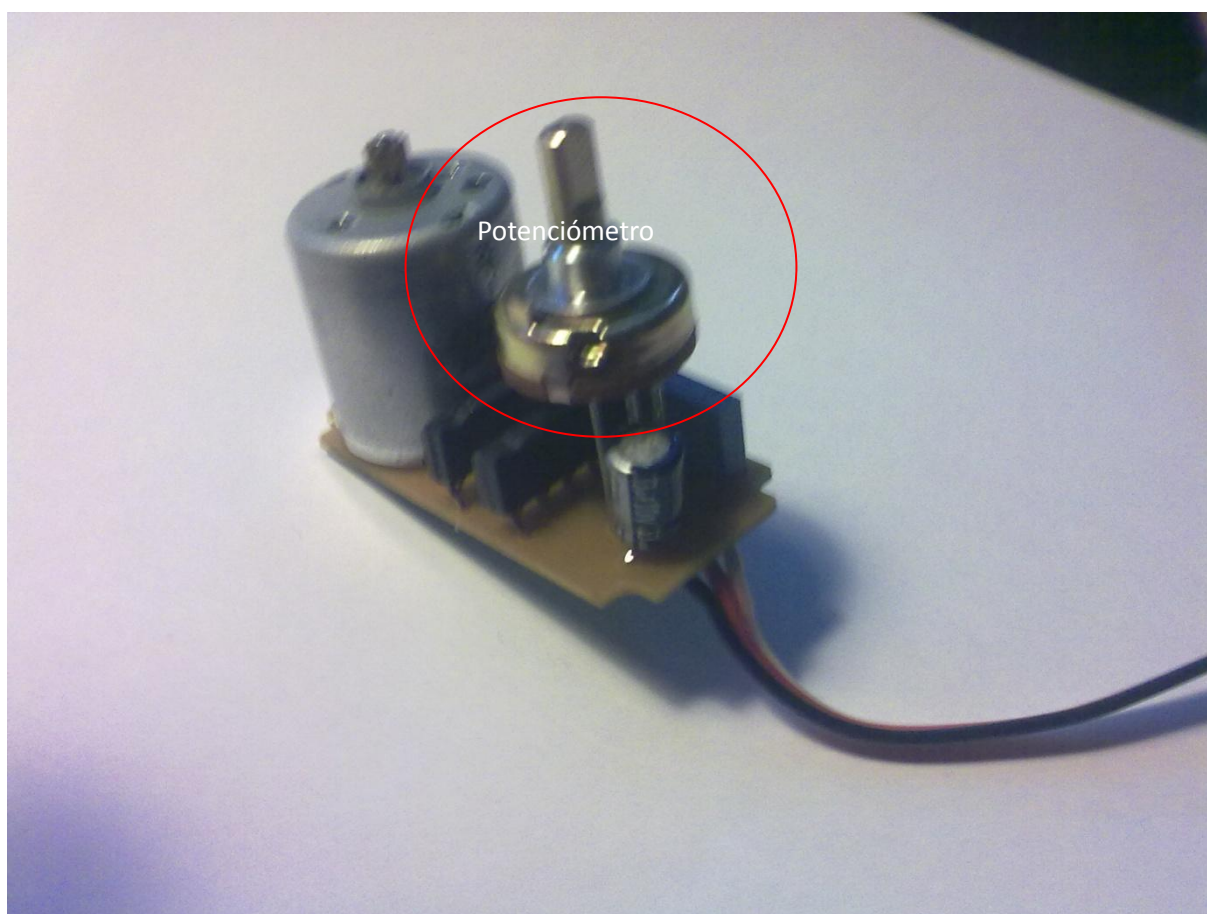


Figura 4: Interior de un servo con el potenciómetro remarcado

### 3.2 Principio De Funcionamiento Del Sistema Sensor

Tal como se ha explicado en el apartado anterior, el propio actuador ya contiene un sensor dentro que indica continuamente la posición a la que se encuentra el servo. Tras un estudio de posibles sistemas sensores, se vio que la mejor forma de conocer la posición del servo era sacando la información del sensor interno del servo. El requisito [R013] lo especifica. Ello conllevaba que no se necesita un sistema sensor complejo mecánicamente. Sin embargo, conlleva el problema de que hay que sacar información directamente del actuador, y por tanto hay que tratar de evitar que nuestro sistema afecte al actuador. El sistema sensor debe ser transparente para el actuador.

El potenciómetro es un tipo de sensor de resistencia variable, que varía mediante un desplazamiento angular. La tensión que hay entre las conexiones del potenciómetro, por tanto, van variando. Esa variación de tensión es la que utiliza el circuito de control para conocer la posición del servo, y es esa misma variación la que nosotros queremos medir.

El funcionamiento de los sistemas de sensor sigue un esquema prácticamente idéntico independientemente de su uso, como detalla la Figura 5.

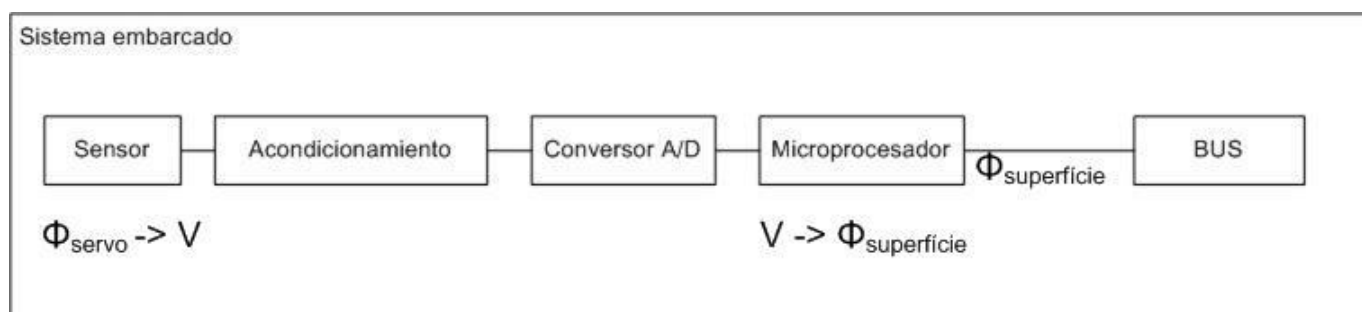


Figura 5: Diagrama de bloques del Sistema Sensor

En el caso que nos ocupa, el acondicionamiento del sensor ya está hecho ya que ya está montado en la propia placa del servo, y por tanto no haría falta realizarlo. Además, los microprocesadores actuales disponen de conversores A/D internos, por lo que no hará falta añadir ninguno. Finalmente, los datos procesados de la señal del sensor deben ser enviados por un BUS de datos.

### 3.3 Elección Del microcontrolador

La elección del microcontrolador es muy importante ya que debe cumplir con ciertos requisitos de diseño. El microcontrolador escogido es el MSP430F2274. La principal razón, además de cumplir con los requisitos de diseño tal y como se explicará más adelante, es porque se comercializa en un módulo de desarrollo, el MSP430 RF2500, que viene incluido en el kit de desarrollo eZ430-RF2500. Este módulo ya desarrollado permite simplificar el diseño pues ya se encuentra el microcontrolador insertado en una placa y no es necesario adaptar ni preparar el sistema. Además, dispone de pins para la conexión y es muy sencillo de grabar el software en él.

Como se ha mencionado, el microcontrolador cumple los requisitos de diseño impuestos. El requisito [R022] de diseño indica que el microcontrolador debe tener un conversor analógico-digital de mínimo 5 bits. El MSP430F2274 tiene dos conversores ADC, uno de 12 bits y otro de 10 bits [3]. Cumple con el requisito. El requisito [R023] y su derivado [R024], indican que el sensor debe estar conectado a un amplificador operacional, y que este debe estar integrado en el microcontrolador. El MSP430F2274 contiene dos amplificadores operacionales y por tanto es suficiente y cumple con los requisitos impuestos. La razón principal de la utilización del amplificador viene explicada en el apartado 6, que detalla el motivo de SAFETY que lleva a la utilización de éste, entre otros aspectos.

Además, el microcontrolador permite la creación de su código mediante el uso de lenguaje de programación C, y permite controlar absolutamente todo el sistema a través de un único software concentrado en él, cumpliendo con el requisito [R027].



### 3.4 Conexión de nuestro sistema sensor al actuador

Para la conexión entre el potenciómetro del servo y nuestro sistema, será necesario hacer una pequeña modificación en el servo. Los servos son fácilmente desmontables ya que no tienen piezas soldadas, es todo mediante uniones mecánicas (tornillos). Será necesario abrir por tanto el servo para poder conectar nuestro sistema.

A los extremos del potenciómetro que están soldados a la placa del servo (Figura 6) habrá que soldar un cable a cada extremo para poder realizar la conexión en paralelo de nuestro sistema.

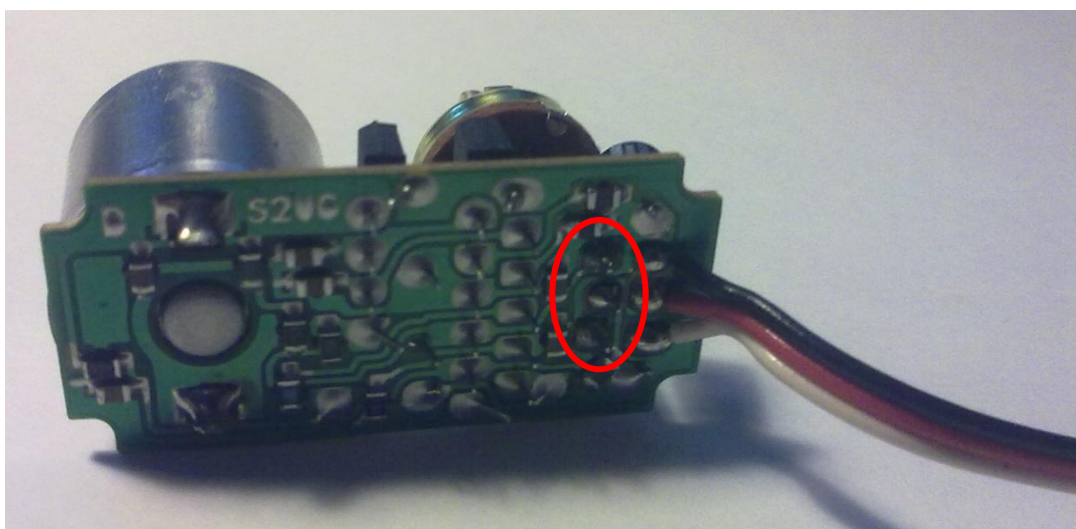


Figura 6: Detalle de la conexión del Potenciómetro al circuito

Esa soldadura deberá hacerse a la conexión central y a la de tierra del potenciómetro.

Para poder sacar esos cables, habrá que hacer unos pequeños agujeros en la tapa del servo (Figura 7). Se harían con un pequeño taladro. Otra opción es utilizar la propia obertura de los cables del servo, agrandándola un poco para permitir el paso de los cables.



Figura 7: Zona a perforar para sacar los cables que van al potenciómetro



En la Figura 8 se puede apreciar un esquema de las conexiones de los cables tras la modificación

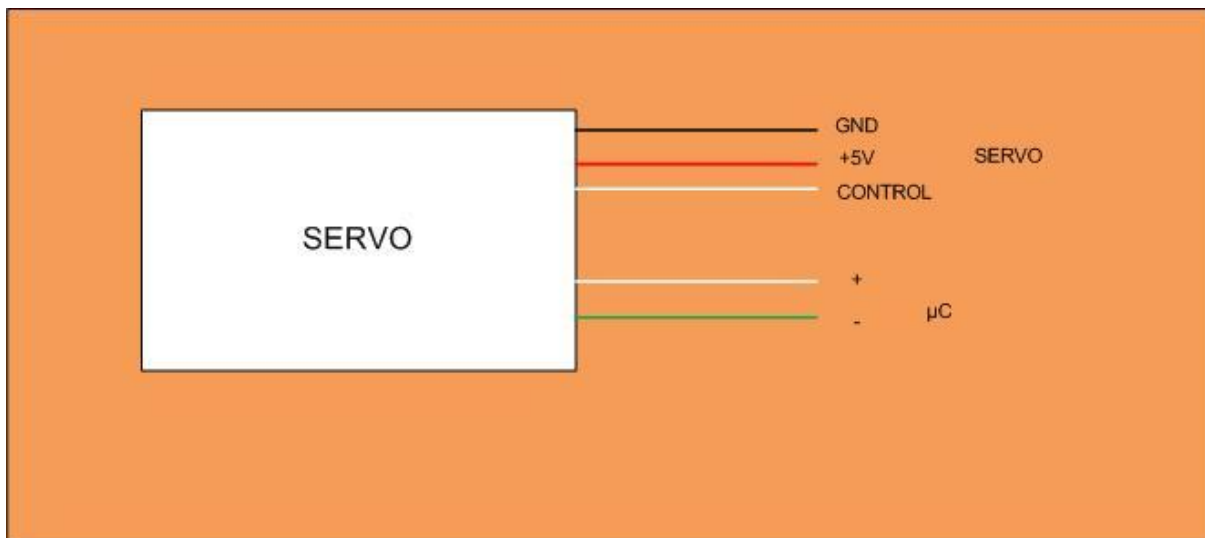


Figura 8: Esquema del servo con la modificación

El resultado se puede ver en la Figura 9, donde ya hay un servo cerrado con los dos cables saliendo de él a través del propio hueco del servo.



Figura 9: Servo con los cables soldados y cerrado

### 3.5 Conexión del potenciómetro al microcontrolador

La conexión al microcontrolador se llevará a cabo a través de alguno de los pins de acceso que tiene el módulo MSP430 RF2500 [4], conectando el cable de polaridad negativa o de tierra que hemos soldado al servo a la conexión de tierra del MSP430. La elección de en qué pin se conectará el potenciómetro se deja a la espera del desarrollo del diseño de software, donde se detallará mejor el porqué de la elección de tal conexión.

### 3.6 Transmisión de datos

Por los requisitos impuestos por el cliente, los datos obtenidos deben ser transmitidos a través tanto de un puerto serie como por Ethernet ([R014] y [R017]). Para ello, el requisito [R025] especifica que el microcontrolador utilizado debe ser capaz de gestionar dos conexiones simultáneas del tipo USCI (interfaz de conexión serie universal). Eso permitirá que se pueda realizar a la vez la conexión serie y Ethernet. El microcontrolador MSP430F2274 dispone de dos módulos USCI, cumpliendo con el [R025]. Se pueden programar para realizar una conexión en SERIE, siguiendo el protocolo establecido para ello. Sin embargo, el módulo no está preparado para emitir a través de una red ETHERNET. La Figura 10 muestra un diagrama de las conexiones USCI.



Figura 10: Diagrama de conexiones USCI del microcontrolador

Para solventar ese problema, se utiliza la placa WIZ810MJ de Wiznet, que incorpora el chip de ETHERNET W5100, y se puede conectar a través de la conexión USCI restante del microcontrolador, utilizando un protocolo SPI. Así, el módulo Ethernet trabaja de forma sincronizada y esclava con el microprocesador, y cumple con el requisito de diseño [R026]. Esta placa no necesita programación, puesto que ya está preparada para enviar por ethernet los datos que le vengan del puerto USCI del microcontrolador. La placa WIZ810MJ tiene un conector RJ-45 hembra al que conectar un cable para transmitir por ethernet. [5]

### 3.7 Alimentación y consumo

En cuanto a la alimentación, el potenciómetro del servo no necesita alimentación ya que coge la propia del servo. Sin embargo, tanto el módulo MSP430 RF2500 como el WIZ810MJ necesitan una tensión de entrada de 3.3 V (de ahí deriva el requisito [R018]). Esa alimentación llegará a través de una línea de transmisión de potencia conectada a las baterías del avión, que emiten 5 V. La Tabla 1 especifica los consumos de los dos componentes principales.

	Consumo máximo (A)
WIZ810MJ [5]	185 mA
MSP430 RF2500 [4]	0.390 mA
Total	185,39 mA

Tabla 1: Consumo del Sistema

Debido a que queremos que tenga 3.3 V de entrada constantes, se utilizará un regulador de tensión, en este caso el MCP1702, un regulador Ldo (low dropout regulator) que funciona con tensiones de entrada y salida muy pequeñas. La utilización de este tipo de regulador viene indicada en el requisito [R020]. Este regulador transforma y mantiene la tensión recibida de la línea de tensión y la deja en la tensión deseada. Debido a los consumos que tiene nuestro sistema, el requisito [R021], derivado del [R020], debe poder aceptar una corriente máxima igual o superior a 200 mA. En nuestro caso, el regulador MCP1702 permite el paso de hasta 250 mA, según indica su datasheet [6].

Los reguladores necesitan un sistema de acondicionamiento formado por condensadores, cuyos valores se pueden encontrar fácilmente en el datasheet del regulador [6]. La Figura 11 muestra el acondicionamiento del regulador LDO.

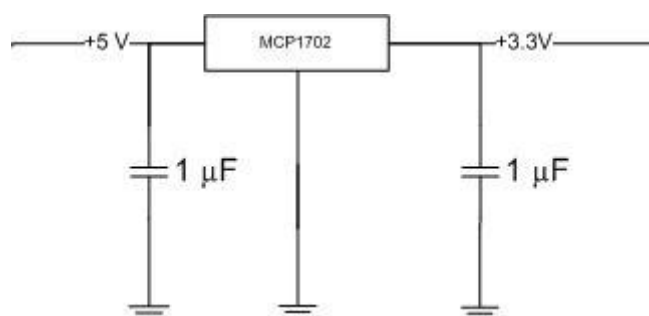


Figura 11: Acondicionamiento del regulador MCP1702

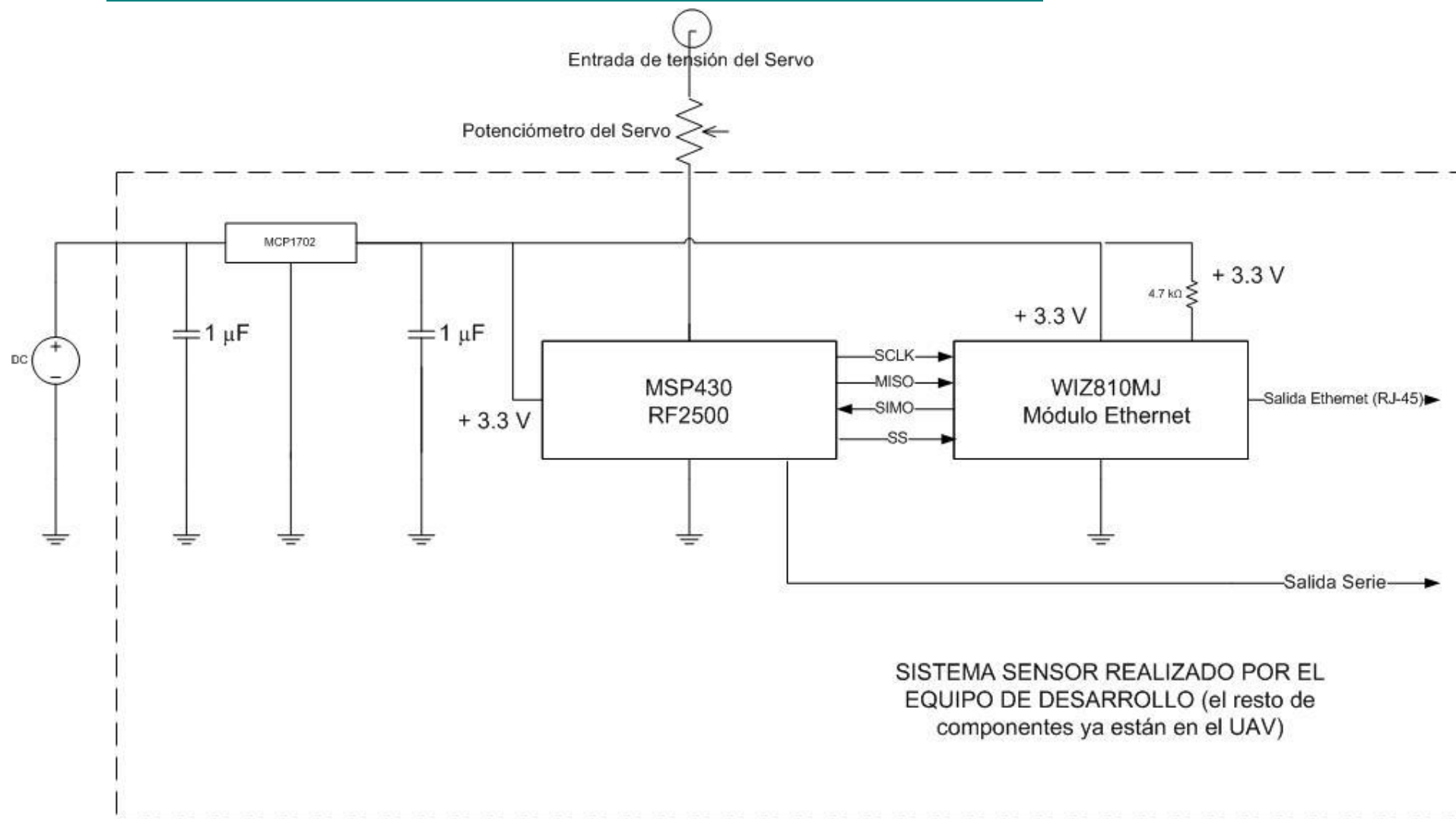


Figura 12: Esquema de componentes del sistema sensor

## 4 DISEÑO DE SOFTWARE

El diseño de software es muy importante ya que es el encargado de controlar todas las partes del sistema y por tanto, se debe realizar controlando cada parámetro para el control de este. Se considera que el software debe estar todo concentrado en el microcontrolador, es decir, no debe necesitarse software para ninguna otra parte del sistema (requisito [R027]). En la Figura 13 vemos un diagrama de la funciones del software. Esta división en funciones viene impuesta por el requisito [R029]. Además, la asignación de cada función viene dada por los requisitos [R030], [R031], [R032], [R033], [R034] y [R035].

Funciones del software:

- Función 1: Entrada de la señal del sensor
- Función 2: Amplificación de la señal
- Función 3: Conversión Analógico a Digital
- Función 4: Tratamiento de datos (obtención de la posición de la superficie)
- Función 5: Envío de datos (SERIE)
- Función 6: Envío de datos (ETHERNET)

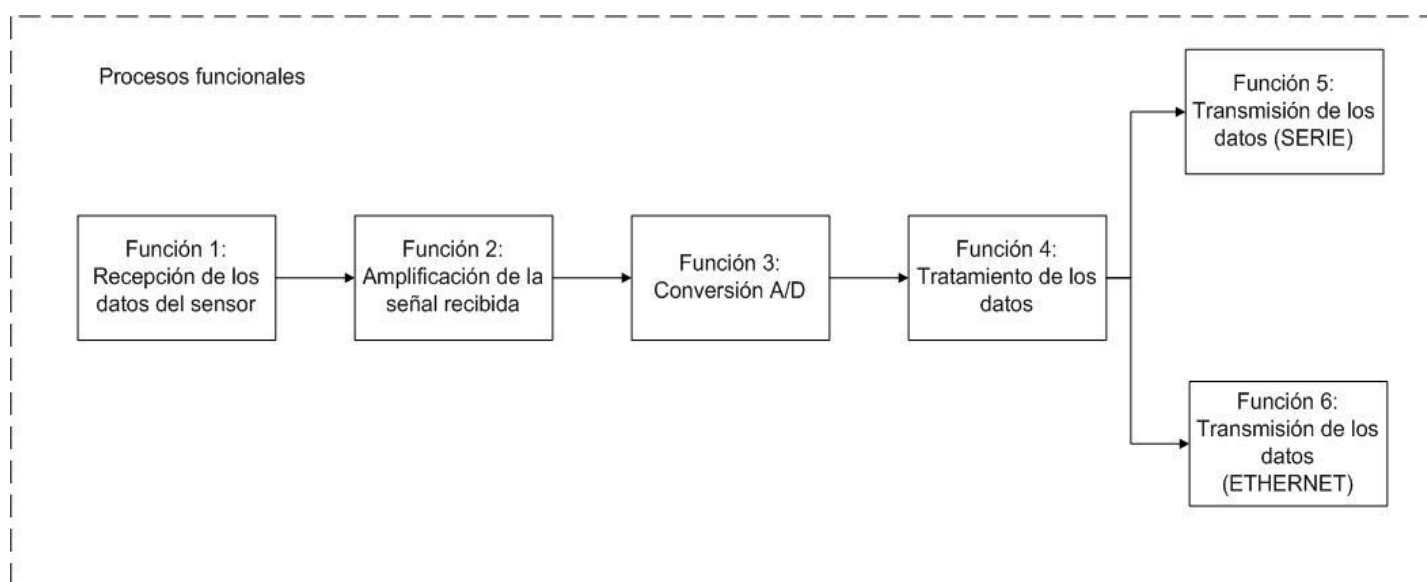


Figura 13: Diagrama de funciones

La función 4 es la única propiamente de software. El resto, son funciones que aunque incluyen cierta parte de software, son mayoritariamente de configuración del sistema.

#### 4.1 Función 1: Recepción de los datos del sensor [R030]

Conexión a la entrada analógica del amplificador operacional I0 del microcontrolador a través del Pin 3 del módulo MSP430 RF2500 [3].

Características del pin 3 del módulo MSP430 RF2500

- Entrada y salida digital (puerto 2, pin 0 del microcontrolador, P2.0)
- Entrada analógica A0, conectada al conversor ADC10
- Salida del reloj auxiliar ACLK
- Entrada analógica I0 del amplificador operacional OA0

#### 4.2 Función 2: Amplificación de la señal recibida [R031]

Conexión mediante la señal que entra a través de la entrada OA0I0. Se utilizará el amplificador operacional OA0 para amplificar la señal. Amplificador OA0 configurado como amplificador No-Inversor, tal y como indica el requisito [R037]. De esta forma, la tensión que llega a la entrada del amplificador se amplifica sin absorber corriente del sistema.

Se utilizará una ganancia  $G=1$ . La explicación de la elección de  $G=1$  se encuentra explicada en la función 3, y deriva del requisito [R036].

La salida se configurará a la entrada A1 del ADC10.

#### 4.3 Función 3: Conversión A/D [R032]

Recepción a través de la entrada A1 del ADC10. ADC10: Conversor A/D de 10 Bits.

Se debe configurar una tensión máxima y una mínima. Cuando la tensión recibida sea mayor o igual a la tensión máxima, la salida digital será de todo el rango (03FFh). La salida digital será 0 cuando la tensión recibida sea menor o igual a la tensión mínima (00000). Para poder indicar al microprocesador las tensiones mínimas y máximas, se puede hacer mediante el uso de tensiones externas o internas. Para no complicar el diseño, se utilizan tensiones internas, que serán  $V_{REF+} = 2.5\text{ V}$  y  $V_{SS} = 0\text{ V}$ , siendo  $V_{REF+}$  la tensión de referencia interna y  $V_{SS}$  la toma de tierra.\*

La formula de conversión será la siguiente

$$N_{ADC} = 1023 \times \frac{V_{in} - V_{SS}}{V_{REF+} - V_{SS}} \text{ cuentas}[3]$$

\*Debido a que la tensión máxima que puede tener el potenciómetro utilizado en el prototipo es de 2.2 V, nos servirá en este caso usar esas tensiones de referencia. Sin embargo, si se conectara otro potenciómetro que funcionara a tensiones mayores de 2.5V, habría que reconfigurar el sistema. Sin embargo, es bastante improbable que tal cosa suceda debido al tipo de potenciómetros utilizados en los servos. La forma más fácil sería utilizar en vez de  $V_{REF+}$  como tensión máxima,  $V_{CC}$ .

El resultado  $N_{ADC}$  se guardará en una variable a la cual se podrá acceder para modificarla y obtener los datos deseados.

#### 4.4 Función 4: Tratamiento de los datos [R033]

Las variables introducidas por el usuario se utilizan para el correcto funcionamiento del sistema. Además, se tiene una variable con los datos de la tensión del potenciómetro en formato digital.

Antes del procesado de datos, es necesario saber que para el correcto funcionamiento del sistema, se tiene que utilizar la variable que determina la posición mínima como mayor que 0. Para eso, se realiza un algoritmo al inicio del programa, tal como detalla la Figura 14:

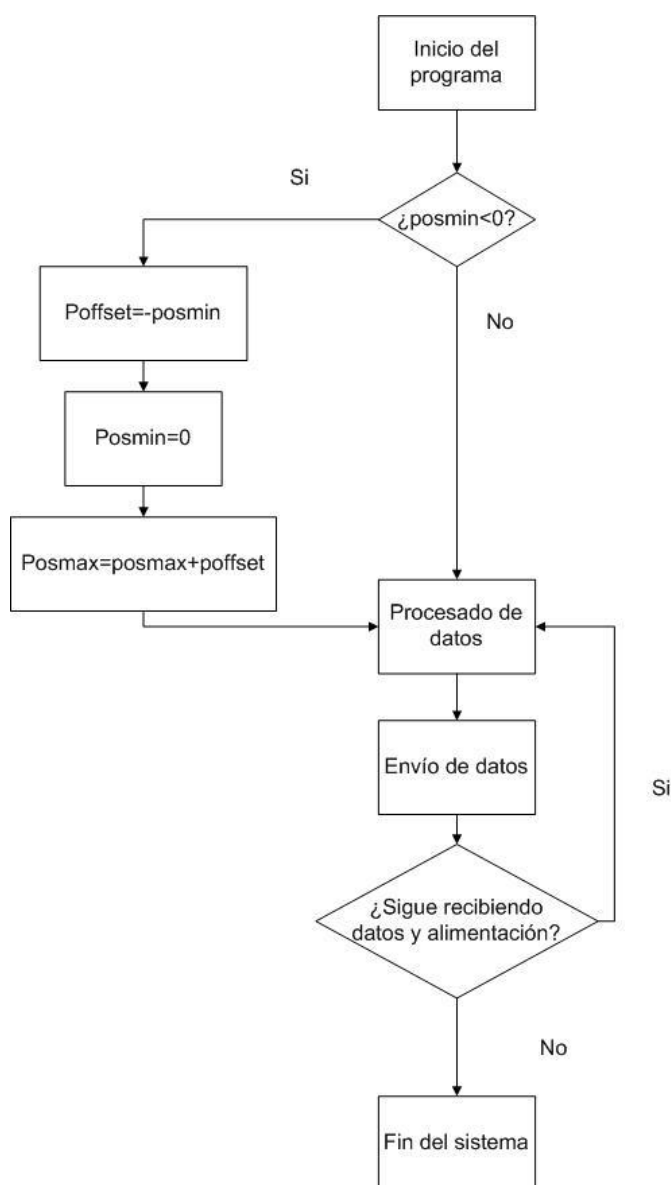


Figura 14: Algoritmo de inicialización de la variable posmin y posmax

El rango dinámico sacado de los requisitos de diseño es  $DR=20$ . Eso nos da 20 posiciones posibles del servo. Gracias a que sabemos la posición mínima y máxima de cada superficie a controlar, no es necesario convertir la tensión del potenciómetro en la posición del servo para luego calcular la posición de la superficie en función de la posición del servo. En vez de eso, pasamos directamente la tensión del servo a la posición de la superficie. Para ello, solo hemos de conocer la posición mínima y máxima de la superficie, y la tensión mínima y máxima convertida a digital que ofrece el potenciómetro del servo.

La tensión máxima que puede tener el potenciómetro es de 2.2 V. Esto en formato digital quiere decir un valor de:

$$N_{adc}=1023*(2.2-0)/(2.5-0)=900* \text{ cuentas}$$

La tensión mínima del potenciómetro es de 0.0748 V. Seguimos el mismo procedimiento para pasarlo a formato digital:

$$N_{adc}=1023*(0.0748-0)/(2.5-0)=30* \text{ cuentas}$$

\* El microprocesador no utiliza decimales para esta operación

Por tanto, el rango en digital será de 870 cuentas.

Para poder asignar a qué posición de la superficie se encuentra, miramos el rango de movimiento de la superficie a medir. El programa estará automatizado para dar la posición de la superficie final deseada. En este caso, la superficie es directamente la posición del servo de pruebas, con una posición angular mínima de  $0^\circ$  y una máxima de  $210^\circ$ . De aquí se puede sacar una ecuación de una recta en función de la tensión.

$$Posición\ servo = (0.2414 * N_{ADC} - 7.2414)^\circ$$

La resolución en grados es de  $10.5^\circ$  (resolución = Rango/DR). Por tanto, la posición del servo será

$$Posición\ servo = (0.2414 * N_{ADC} - 7.2414)^\circ \pm 10.5^\circ$$

Esta será la información a enviar.

Para facilitar la adaptación del prototipo a cualquier sistema, los datos que se tienen que insertar serán:

- Posición mínima de la superficie (debe ser mayor o igual a 0)
- Posición máxima de la superficie
- Tensión mínima que pasa por el potenciómetro del servo (en milivoltios)
- Tensión máxima que pasa por el potenciómetro del servo (en milivoltios)
- Identificador de la superficie



A partir de los datos obtenidos, se transmite la información mediante un mensaje de 11 Bytes. Este mensaje contiene una cabecera (SURF), un número identificador de la superficie formado por dos números y el valor numérico de la posición de la superficie monitorizada precedido de su signo en función de que sea positivo o negativo. El mensaje tiene el formato especificado en el requisito de diseño [R039], que especifica el envío del mensaje con una cabecera, un número de identificador y la posición del sistema. El mensaje enviado será:

"SURFXX.±YYY",

Donde XX es el número que identifica a la superficie que se está monitorizando e YYY es el valor numérico de la posición de la superficie (capaz de transmitir hasta centenas). Si el sistema detectara que ha habido un error en los valores (no son coherentes con lo que debería dar), en la posición donde debería ir la información de la posición las letras ERR, que indicarían que ha habido un error. Quedaría un mensaje como el siguiente:

"SURFXX. ±ERR".

Este mensaje responde al requisito [R040].

#### 4.5 Función 5: Transmisión de los datos en serie [R034]

La transmisión de datos mediante la conexión serie se realizará utilizando la configuración de la tabla 1:

Bits por Segundo	9600
Bits de Datos	8
Paridad	Ninguna
Bits de Parada	1
Reloj	8 MHz

Tabla 2: Propiedades de la conexión SERIE

Esta configuración se escoge debido a la sencillez del sistema realizado y del mensaje enviado. Se podría utilizar una velocidad de transmisión mayor, sin embargo, eso aumentaría el consumo de recursos del sistema y además, el puerto serie solo se utilizará como puerto de pruebas, ya que la transmisión se hará principalmente a través de la red Ethernet.

#### 4.6 Función 6: Transmisión de los datos por ETHERNET [R035]

El cliente pidió que el sistema pudiera emitir por ETHERNET ya que se espera que en un futuro se utilice la red MAREA en el UAV como sistema de BUS. Tras hablar con el equipo encargado del sistema MAREA, se añaden los requisitos [R041] y su derivado [R042]. Los requisitos se imponen ya que la mejor opción contemplada será emitir el mensaje de información a través de toda la red y mandarlo a un puerto conocido, para que pueda recoger la información el usuario que lo desee, utilizando un protocolo UDP. Sin embargo, si se deseara cambiar la información del receptor, el sistema estará preparado para cambiar la dirección del receptor. En la Figura 15 hay un esquema de la distribución de la red y las variables necesarias para transmitir la información.

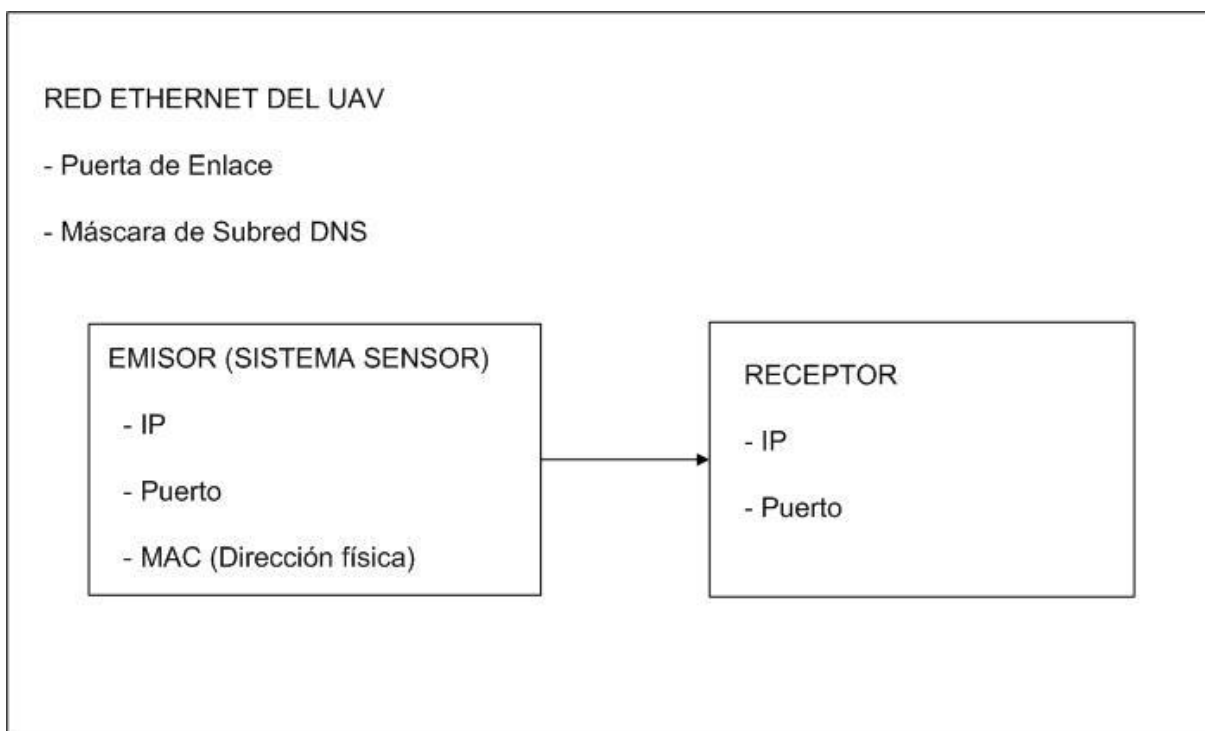


Figura 15: Esquema de la red y relación con nuestro sistema

La transmisión se realiza a través de un componente externo del microcontrolador, el módulo de Ethernet WIZ810MJ que incorpora el chip W5100. Es un componente que solo precisa la programación del socket de información que se utilizará, sin necesidad de configurar los protocolos de transmisión. Este módulo funcionará de forma esclava (SPI) con el microcontrolador utilizado por el sistema, de forma que estén sincronizados.

## 4.7 Distribución de los ficheros que forman el software

El sistema está formado por 4 ficheros que componen todo el software necesario. De esos ficheros, 3 son cabeceras que contienen variables, y uno es el archivo de código. La Figura 16 muestra un esquema de la distribución de los ficheros, indicando los ficheros que tienen variables para cambiar y los que no.

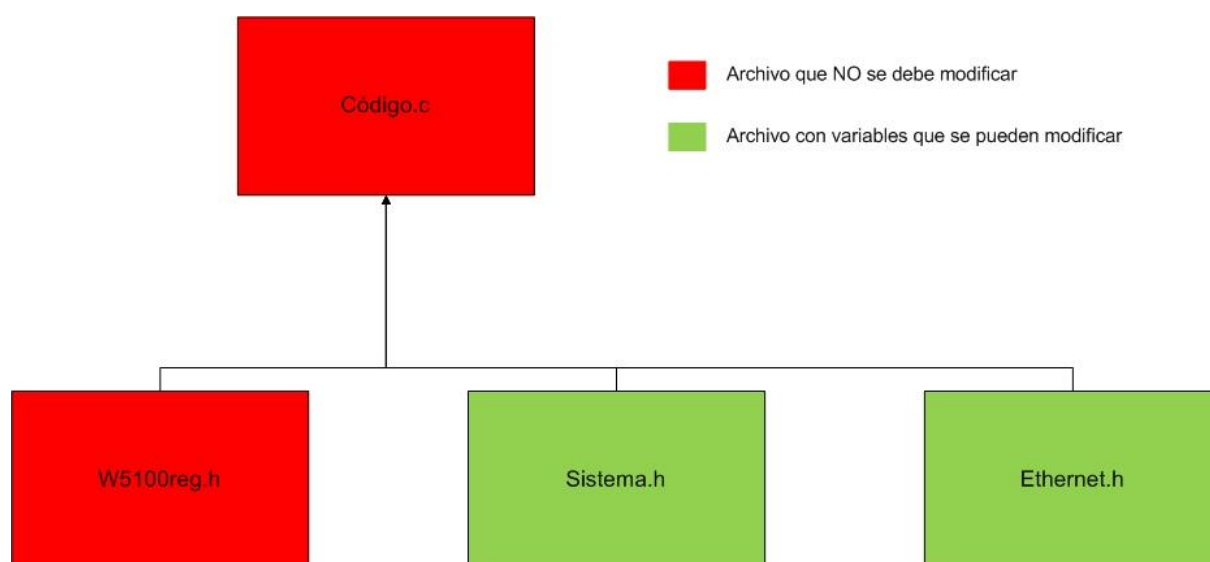


Figura 16: Distribución de ficheros de software

El archivo código.c contiene la base del programa, las funciones y todos los procesos que controlan el sistema. No hay que modificar nada en este archivo si se quiere asegurar el correcto funcionamiento del sistema. Cualquier cambio en este no está contemplado y por tanto solo debe tocarse si se conoce con seguridad lo que se está cambiando en este.

El archivo w5100reg.h contiene las direcciones de los registros de configuración con los que funciona el módulo WIZ810MJ con el chip de Ethernet W5100 y los registros de control del SOCKET 0. No es necesario cambiar nada en este archivo. Si se deseara utilizar alguno de los otros 3 SOCKETS restantes se podrían incluir en este archivo los registros de control de esos SOCKETS (NO ES NECESARIO PARA EL FUNCIONAMIENTO DEL SISTEMA).

Los archivos sistema.h y Ethernet.h son los archivos que contienen las variables que se deben modificar para adaptar el sistema a otro sistema externo. Es importante remarcar que los datos tienen que ser números enteros, ya que el microcontrolador no está pensado para trabajar con decimales. Para más información, revisar el manual del sistema para la modificación de variables [13].



## Diseño de Sistema Sensor Para superficies de UAV

Ref.:  
Iss./Rev.: D03  
Date: 20/06/2009

La elección de esta estructura representa un esfuerzo en facilitar la adaptación del sistema a cualquier superficie ya que solo necesita el cambio de unas variables en los archivos de cabecera. Esto se ha desarrollado de este modo para cumplir los requisitos [R028] que exigen sencillez para la adaptación del sistema a cualquier superficie, y el requisito derivado [R038], que especifica el uso de ficheros de cabeceras para la modificación de las variables de la superficie.

## 5 CONEXIONES DEL SISTEMA

El sistema necesitará poder conectarse entre las diferentes partes para poder funcionar correctamente. Debido a la relativa sencillez del sistema y el requisito de diseño [R019], se ha desarrollado una placa de circuito impreso en la cual se soldarán todos los componentes. La realización de esta placa se ha llevado a cabo utilizando el programa *FREEPCB* de libre distribución y con los servicios de laboratorio de la EPSC para la creación física. Tras tener todos los componentes físicos, se realiza un diseño utilizando las diferentes medidas de cada subsistema, con las conexiones detalladas en la Figura 17.

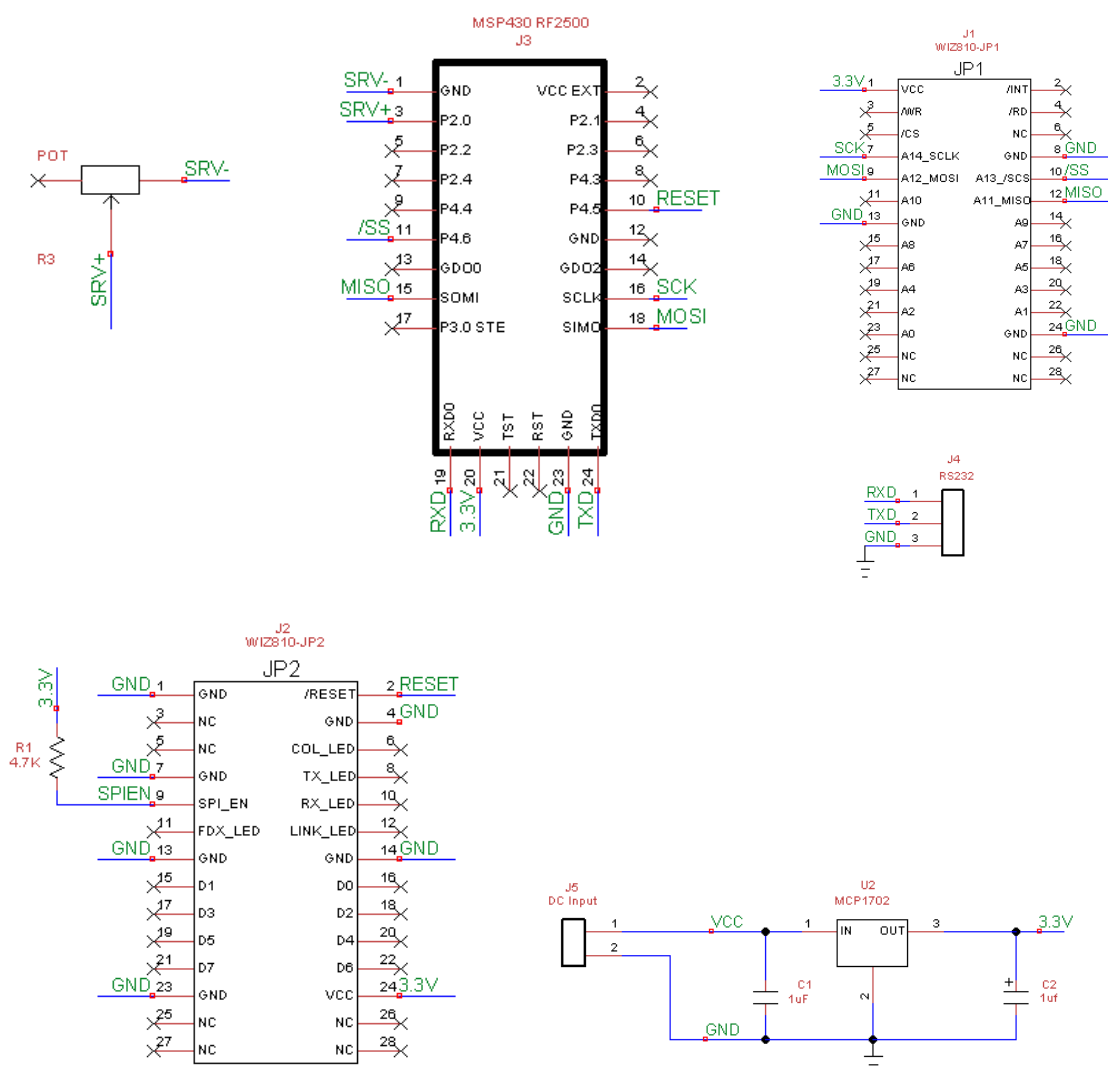


Figura 17: Esquema de conexiones del sistema

De estas conexiones se realiza la placa para el prototipo. Es una primera placa que se ha diseñado para un prototipo inicial, pero cuyas medidas (156x56 mm) pueden verse fácilmente reducidas con un rediseño de esta placa, utilizando el esquema de conexiones anterior. La imagen del diseño de la placa se puede ver en la Figura 18.

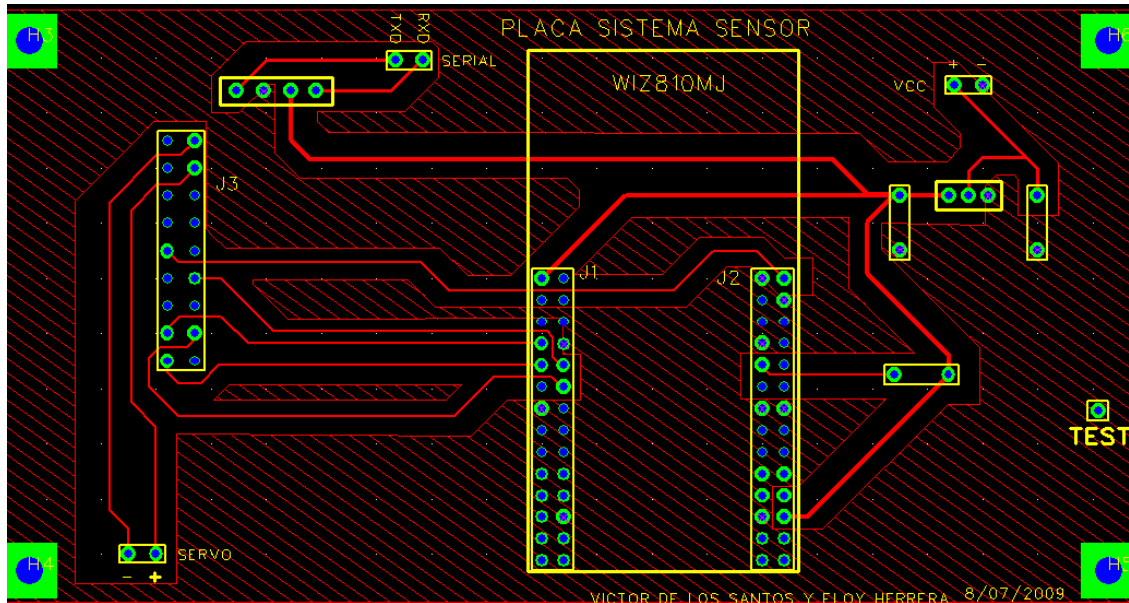


Figura 18: Esquema de la PCB utilizada para el prototipo

Se le han añadido entradas para conectores de polaridad para recibir la señal de tensión, la transmisión serie y la conexión al servo. El resultado final de esta placa una vez ensamblada puede verse en la Figura 19.

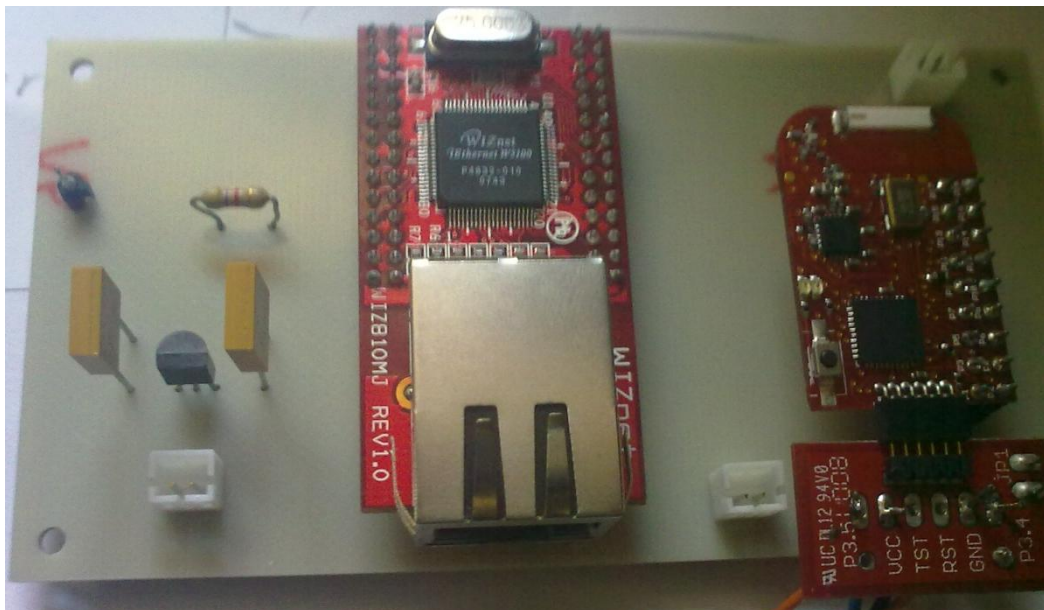


Figura 19: PCB ensamblada

## 6 RELACIÓN DEL DISEÑO CON LOS ASPECTOS DE SAFETY

Durante el desarrollo del FHA y el PSSA, se considera que el sistema embarcado es un sistema que tiene un DAL de tipo C (“major condition failure”). Todas sus partes tienen asignadas el mismo DAL, debido a que el sistema no cuenta con redundancias, y por consiguiente, un fallo en una de las partes provocará un fallo en todo el sistema.

### 6.1 Requisitos del Preliminary System Safety Assessment

En la elaboración del PSSA es donde se desarrollan todos los requisitos DAL C, que no podremos cumplir ya que los componentes utilizados no disponen de MTBFs porque no son de uso aeronáutico. Sin embargo, se tiene en cuenta ciertos requisitos añadidos.

**Requisito [RS035]:** La señal no se verá afectada por el ruido generado en el avión. Este requisito lo cumplimos aplicando ciertas medidas importantes. La primera, es alejar nuestro sistema de los cables de alimentación y motores lo máximo posible, tal y como está detallado en el diseño arquitectural. La localización se realiza en esa zona de forma que se atenúe lo máximo posible la señal de ruido. El ruido máximo que obtenemos, sin embargo, será el ruido que tenga el potenciómetro del servo, y ese ruido ya no podemos controlarlo nosotros, porque, aunque se podría filtrar, el circuito de control recibe la tensión con el mismo ruido.

**Requisito [RS036]:** El sistema no afectará el correcto comportamiento mecánico de las superficies de control. Para realizar esto, se colocan los cables y el sistema, al igual que en el requisito [RS035], lo más alejado del servo posible. Además se comprobará que ningún cable ni conexión dificulta el movimiento de las superficies.

**Requisito [RS037]:** El sistema no interferirá en la aerodinámica de la aeronave. Una de las razones principales por las que se decidió realizar este diseño, se basa en este requisito. Otros diseños conllevaban colocar piezas en el exterior de la aeronave, con lo que la aerodinámica del sistema variaría, y podría conllevar graves problemas. Por eso, se realizó un diseño que se concretará todo dentro del avión, de forma que la aerodinámica no se viera comprometida.

### 6.2 Requisitos del Zonal Safety Assessment

En la elaboración del ZSA nos damos cuenta que al soldar dos cables al potenciómetro para tomar una medida de la resistencia de éste, estamos tocando un subsistema perteneciente a la función de control de la aeronave. Por lo que según el documento NASA/TM-2007-214539 nuestro sistema pasaría de tener una “major condition failure” (Conclusión extraída del PSSA) a “catastrophic condition failure”. Esto ahora implica que todos los subsistemas pasarán a tener un **DAL A**. La máxima prioridad ahora será no falsear la medida que el sistema de control propio del servomotor toma del potenciómetro. En la Figura 20 se ve remarcada en la tabla de funcionalidades críticas de la aeronave en que parte del avión se encuentran las superficies de control.

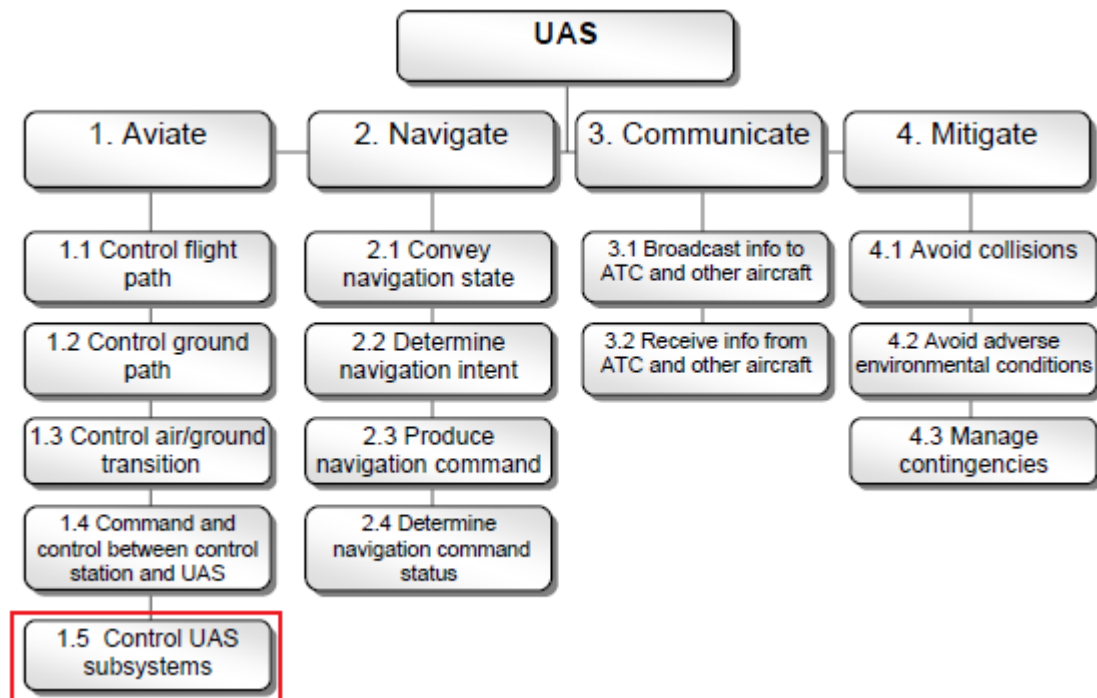


Figura 20: Funcionalidades críticas de la aeronave

El potenciómetro del servo está conectado en paralelo a nuestro sistema. Eso provoca que la tensión que reciba el sistema sea la misma que recibe el circuito de control del servo. Sin embargo, las conexiones en paralelo hacen que la corriente se divida en las dos conexiones. Para evitar esto, se utiliza un amplificador operacional no inversor ya incluido en el módulo MSP430 RF2500. Los amplificadores operacionales amplifican la señal de tensión pero la entrada de corriente es igual a cero (tienen una impedancia muy grande). La Figura 21 muestra las tensiones y corrientes de entrada de un amplificador operacional.

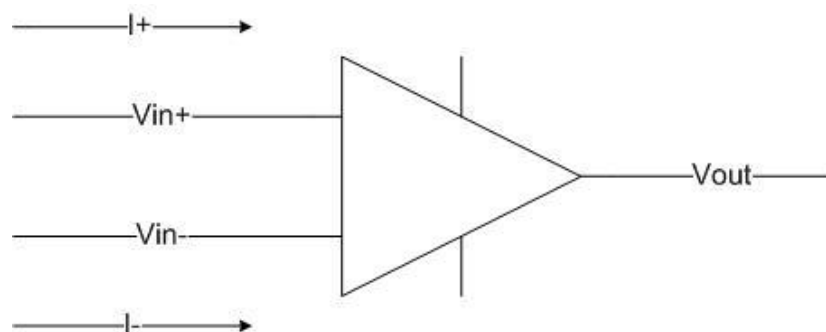


Figura 21: Esquema de un amplificador

$$V_{out} = G * (V_{in+} - V_{in-})$$

$$I+ = I- = 0 \text{ A}$$





## Diseño de Sistema Sensor Para superficies de UAV

Ref.:  
Iss./Rev.: D03  
Date: 20/06/2009

Esto conseguirá así satisfacer los requisitos del Zonal Safety Assessment realizado. Se contempla que, debido a un fallo en el microcontrolador, que por defecto tiene todas las conexiones configuradas como entradas, pueda empezar a emitir por lo que es la entrada en dirección hacia el servo. Esto es altamente improbable, pero existe una pequeña posibilidad de que suceda por causas ajenas al software (ya que está configurado para tener la conexión EXCLUSIVAMENTE como entrada).



## 7 REFERENCIAS

- [1] *Servomotores*, Yamid Ramirez, <http://www.monografias.com/trabajos60/servo-motores/servo-motores.shtml>, última fecha de revisión: 11/07/09
- [2] ICARUS TEAM, *SHADOW MK-1 FLIGHT MANUAL*. 19/05/2009, CASTELLDEFELS
- [3] MSP430 2274 Family User's Guide, 20/06/09, <http://focus.ti.com/lit/ug/slau144e/slau144e.pdf>
- [4] MSP430 RF2500 Development Kit, 20/06/09, <http://focus.ti.com/lit/ug/slau227e/slau227e.pdf>
- [5] WIZ810MJ Datasheet, 20/06/09,  
[http://www.circuitcellar.com/wiznet/WIZ810MJ%20Datasheet\\_V\\_1.0.pdf](http://www.circuitcellar.com/wiznet/WIZ810MJ%20Datasheet_V_1.0.pdf)
- [6] MCP1702 Datasheet, 30/06/09,  
<http://ww1.microchip.com/downloads/en/DeviceDoc/21983A.pdf>
- [7] Juan López, Pablo Royo, Cristina Barrado, Enric Pastor, *MODULAR AVIONICS FOR SEAMLESS RECONFIGURABLE UAS MISSIONS*, 30/10/08, CASTELLDEFELS

## 8 GLOSARIO

**UAV:** Unmanned Aerial Vehicle

**USCI** = Universal Serial Communication interface (Interfaz universal de comunicación serie)

**LDO:** Low Drop Out (baja caída)

**TCP:** Transmission Control Protocol (protocolo de control de transmisión)

**UDP:** User Datagram Protocol (Protocolo de datagramas de usuarios)

**ICMP:** Internet Control Message Protocol (Protocolo de mensaje de control de internet)

**IPv4:** Internet protocol version 4

**ARP:** Address Resolution Protocol (Protocolo de resolución de dirección)

**IGMP:** Internet Group Management Protocol (Protocolo de gestión de grupo de internet)

**PPPoE:** Point-to-Point Protocol over Ethernet (Protocolo punto a punto sobre internet)

**ADC:** Analog to Digital Converter (Conversor analógico digital)

**SPI:** Serial Peripheral Interface (Interfaz de periféricos en serie)

**MAC:** Media Access Control address (dirección de control de acceso al medio)

**DAL:** Design Assurance Level ( Nivel de seguridad de diseño)



# Diseño de Sistema Sensor Para superficies de UAV

Ref.:  
Iss./Rev.: D03  
Date: 20/06/2009

[END OF DOCUMENT]

## ANEXO 4: Manual del Sistema

	Name	Signature	Date
Proyectista 1	Eloy Herrera		07/07/09
Proyectista 2	Víctor de los Santos		07/07/09
Director	Jorge Ramírez Alcantara		07/07/09



## Contenido

1	Introducción.....	6
2	OBJETIVO .....	7
3	SUMARIO.....	8
4	MANUAL DE INSTALACIÓN .....	9
4.1	Preparación del hardware.....	9
4.1.1	Contenido del Kit de desarrollo eZ430 RF2500.....	10
4.1.2	Placa PCB .....	12
4.1.3	Montaje del hardware .....	13
4.2	Preparación del Software .....	22
4.2.1	Archivos de código:.....	23
4.2.2	Compilación y grabación del código en el microcontrolador MSP430.....	28
5	MANUAL DE CALIBRACIÓN .....	34
5.1	Opción 1: .....	34
5.2	Opción 2:.....	34
6	MANUAL DE USUARIO.....	38
6.1	Mensaje transmitido.....	38
6.2	Recepción de datos por la conexión SERIE .....	38
6.2.1	EJEMPLO: HyperTerminal Private Edition .....	39
6.3	Recepción de datos por la conexión ETHERNET .....	42
6.4	I.C.D.....	42
7	BIBLIOGRAFIA .....	43
8	GLOSARIO.....	44



# Diseño de Sistema Sensor Para superficies de UAV

Ref.:  
Iss./Rev.: D01  
Date: 09/07/2009

## Índice de Tablas

TABLA 1: LISTA DE MATERIAL NECESARIO.....	8
TABLA 2: HARDWARE NECESARIO PARA LA INSTALACIÓN.....	9
TABLA 3: HERRAMIENTAS PARA EL MONTAJE DEL HARDWARE .....	9
TABLA 4: CONTENIDO DEL KIT EZ430 RF2500 .....	10
TABLA 5: VARIABLES DEL ARCHIVO SISTEMA.H.....	24
TABLA 6: INFORMACIÓN DE LA CONEXIÓN SERIE DEL SISTEMA.....	39
TABLA 7: CONFIGURACIÓN DEL ETHERNET DEL SISTEMA POR DEFECTO .....	42
TABLA 8: ICD .....	42

## Índice de Figuras

FIGURA 1: DIAGRAMA DE PARTES DEL HARDWARE	10
FIGURA 2: MÓDULO MSP430 RF2500	10
FIGURA 3: GRABADOR USB	11
FIGURA 4: KIT DE BATERÍAS	11
FIGURA 5: ESQUEMA DE CONEXIONES DEL SISTEMA	12
FIGURA 6: IMAGEN DE LA PARTE INFERIOR DE LA PCB	13
FIGURA 7: CABLES SOLDADOS AL POTENCIÓMETRO	14
FIGURA 8: SERVO CERRADO CON LOS CABLES EXTRAÍDOS	14
FIGURA 9: CONECTOR DE POLARIDAD HEMBRA DEL SERVO	15
FIGURA 10: MÓDULO MSP430 RF2500 CON EL CONECTOR SOLDADO	15
FIGURA 11: CONEXIONES DE LA PCB	16
FIGURA 12: CONECTORES DE POLARIDAD SOLDADOS A LA PCB	17
FIGURA 15: CONDENSADOR DE 1 $\mu$ F	17
FIGURA 13: REGULADOR LDO DE 3.3 V	17
FIGURA 14: RESISTENCIA DE 4.7 K $\Omega$	17
FIGURA 16: REGULADOR, CONDENSADORES Y RESISTENCIAS SOLDADOS A LA PLACA	18
FIGURA 17: PLACA CON LOS PASOS SOLDADOS	19
FIGURA 18: MÓDULOS INSTALADO SOBRE LA PLACA	19
FIGURA 19: PUNTOS A DESOLDAR DEL MODULO DE BATERÍAS	20
FIGURA 20: PLACA DE LA BATERÍA DESOLDADA Y CON LOS CABLES	20
FIGURA 21: PLACA CON TODOS LOS COMPONENTES SOLDADOS Y EL SERVO CONECTADO	21
FIGURA 22: DIAGRAMA DE PREPARACIÓN DEL SOFTWARE PARA EL FUNCIONAMIENTO DLE SISTEMA	22
FIGURA 23: DIAGRAMA DE DISTRIBUCIÓN DE LOS ARCHIVOS	23
FIGURA 24: DIAGRAMA DE LA RED ETHERNET	25
FIGURA 25: DIAGRAMA DE INSTALACIÓN Y PREPARACIÓN DEL COMPILADOR Y GRABADOR	28
FIGURA 26: ICONO DEL PROGRAMA	28
FIGURA 27: VENTANA DE INICIALIZACIÓN DEL IAR EMBEDDED WORKBENCH	29
FIGURA 28: CREACIÓN DE NUEVO PROYECTO DEL IAR EMBEDDED WORKBENCH	29
FIGURA 29: AÑADIR FICHEROS AL PROYECTO	30
FIGURA 30: PROYECTO CON LOS FICHEROS DE CÓDIGO AÑADIDOS	30
FIGURA 31: OPCIONES DE PROYECTO	31
FIGURA 32: GRABADOR USB CONECTADO AL MÓDULO MSP430 RF2500 Y A UN CONECTOR USB DEL ORDENADOR	32
FIGURA 33: PANTALLA DE DEBUGACIÓN E ICONO DE PARADA DE DEBUGACIÓN	33
FIGURA 34: DIAGRAMA DE OBTENCIÓN DE LAS TENSIONES MÁXIMAS Y MÍNIMAS	35
FIGURA 35: MÓDULO MSP430 RF2500 DE CALIBRACIÓN	35
FIGURA 36: GRABADOR USB CONECTADO AL MÓDULO MSP430 RF2500 Y A UN CONECTOR USB DEL ORDENADOR	36
FIGURA 37: CAPTURAR TEXTO CON HYPERTERMINAL	36
FIGURA 38: DIAGRAMA DE OBTENCIÓN DE TENSIONES	37
FIGURA 39: CONEXIÓN POR SERIE AL SISTEMA	38





# Diseño de Sistema Sensor Para superficies de UAV

Ref.:  
Iss./Rev.: D01  
Date: 09/07/2009

---

FIGURA 40: ICONO DEL PROGRAMA HYPERTERMINAL PRIVATE EDITION	39
FIGURA 41: PANTALLA DE INICIALIZACIÓN DE LA CONEXIÓN	40
FIGURA 42: SELECCIÓN DEL PUERTO DE ENTRADA DE LA CONEXIÓN	40
FIGURA 43: CONFIGURACIÓN DEL PUERTO SERIE	41
FIGURA 44: RECEPCIÓN DE DATOS DEL SISTEMA SENSOR MEDIANTE LA CONEXIÓN SERIE	41



## 1 INTRODUCCIÓN

El sistema desarrollado sirve para comprobar la posición de cualquier superficie que esté accionada mediante servos. Este documento explica la preparación y uso tanto del hardware como del software para adaptar el sistema a cualquier superficie. A lo largo del documento se encontrarán referencias, diagramas, explicaciones y fotografías para adaptar el sistema de la forma más sencilla y fácil para que el ingeniero que esté llevando a cabo la preparación no tenga ninguna dificultad a la hora de preparar el sistema.



## 2 OBJETIVO

Crear un manual de instalación que indique el procedimiento a seguir para la preparación del sistema.

Crear un manual de calibración que indique el procedimiento a seguir para calibrar el sistema.

Crear un manual de utilización que indique la forma de utilizar el sistema.

### 3 SUMARIO

Este documento está pensado para ser la única documentación necesaria para poder preparar el sistema para monitorizar cualquier superficie de vuelo. Será necesario tener preparado antes todo el material, tanto de hardware como de software, para no tener problemas de falta de material a lo largo del documento. La Tabla 1 es una lista del material utilizado y que permite comprobar que se tienen todos los componentes.

Material	Cantidad	OK
<b>HARDWARE</b>		
Kit de desarrollo eZ430 RF2500	1	
Módulo Ethernet WIZ810MJ	1	
Placa PCB	1	
Conector de polaridad de paso 2.5 mm	3	
Condensador cerámico de 1 $\mu$ F	2	
Regulador LDO MCP1702	1	
Resistencia 4.7 k $\Omega$	1	
Conector macho-hembra de de paso 2.54 mm de 18x2 pins	1	
Conector macho-macho de paso 2.54 mm de 18x2 pins	1	
Conector macho-hembra de paso 2 mm de 32x2 pins	1	
Cable Rígido unifilar de 0.9 mm	1 m.	
<b>SOFTWARE</b>		
Código.c		
Sistema.h		
Ethernet.h		
W5100reg.h		
Calibracion.h		
IAR Embedded Workbench Kickstart Edition v4.2		
HyperTerminal		
PCB.fpc		
<b>HERRAMIENTAS</b>		
Soldador de hierro para estaño		
Cable de estaño para soldar (60% estaño + 40% plomo)		
Multímetro		
Alicates/pelacables		
Destornillador de estrella		

Tabla 1: Lista de material necesario

El documento se divide en 3 manuales: Manual de Instalación, Manual de Calibración y Manual de Uso. Cada manual contiene una descripción detallada del proceso a llevar a cabo tanto a nivel de software como de hardware para que el sistema funcione correctamente.

## 4 MANUAL DE INSTALACIÓN

El manual de instalación detalla el proceso de montaje y preparación del sistema para poder funcionar con cualquier superficie que funcione mediante servos. NO se detalla el montaje para cada superficie en cuanto a su alojamiento en ella, debido a que no se puede realizar un único sistema que encaje en todas las superficies. Esa tarea será para el ingeniero que esté preparando el sistema.

### 4.1 Preparación del hardware

Para la preparación del hardware será necesaria una serie de componentes que se detallan en la Tabla 2. Además, se recomienda una serie de herramientas que aparecen en la Tabla 3.

HARDWARE	
Kit de desarrollo eZ430 RF2500	1
Módulo Ethernet WIZ810MJ	1
Placa PCB	1
Conector de polaridad de paso 2.5 mm	3
Condensador cerámico de 1 $\mu$ F	2
Regulador LDO MCP1702	1
Resistencia 4.7 k $\Omega$	1
Conector macho-hembra de de paso 2.54 mm de 18x2 pins	1
Conector macho-macho de paso 2.54 mm de 18x2 pins	1
Conector macho-hembra de paso 2 mm de 32x2 pins	1
Cable Rígido unifilar de 0.9 mm	1 m.

Tabla 2: Hardware necesario para la instalación

HERRAMIENTAS
Soldador de hierro para estaño
Cable de estaño para soldar (60% estaño + 40% plomo)
Multímetro
Alicates/pelacables
Destornillador de estrella

Tabla 3: Herramientas para el montaje del Hardware

El Hardware se compone de 4 partes principales, siendo las más importantes el Módulo que contiene el microcontrolador (que como veremos en el apartado 4.2 se incluye en el kit de desarrollo eZ430 RF2500). La relación entre las 4 partes puede verse en la Figura 1.

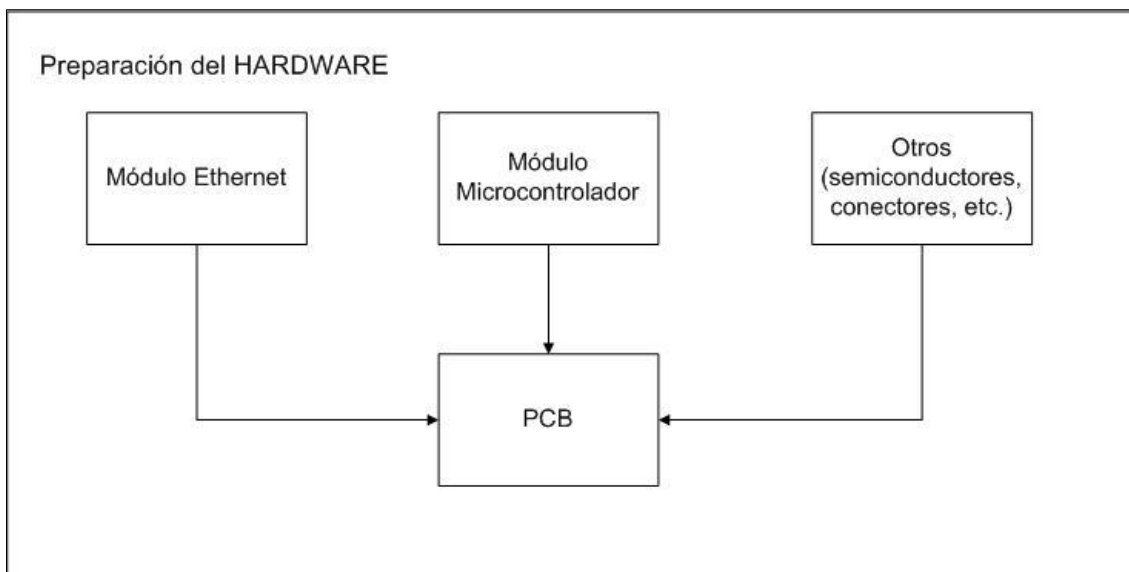


Figura 1: Diagrama de partes del Hardware

#### 4.1.1 Contenido del Kit de desarrollo eZ430 RF2500

El Kit de desarrollo eZ430-RF2500 es un kit que contiene diversos elementos que se van a utilizar. En la Tabla 4 se puede ver el contenido del sistema.

Elemento	Cantidad
Módulo MSP430 RF2500	2
Grabador USB	1
Kit de baterías	1

Tabla 4: Contenido del kit eZ430 RF2500

El módulo MSP430 RF2500, que aparece en la Figura 2, contiene el microcontrolador y las conexiones externas para las entradas y salidas de éste. En el kit de desarrollo se incluyen dos módulos como este.

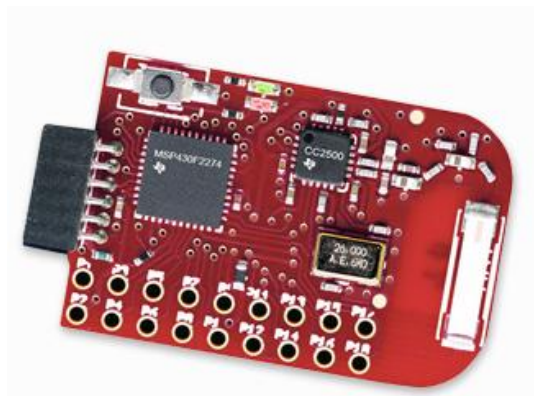


Figura 2: Módulo MSP430 RF2500

El grabador USB es un conector USB que permite conectar el modulo MSP430 RF2500 con el ordenador para grabar el software de control. Además, permite crear una conexión serie entre el ordenador y el módulo. El grabador puede verse en la Figura 3.

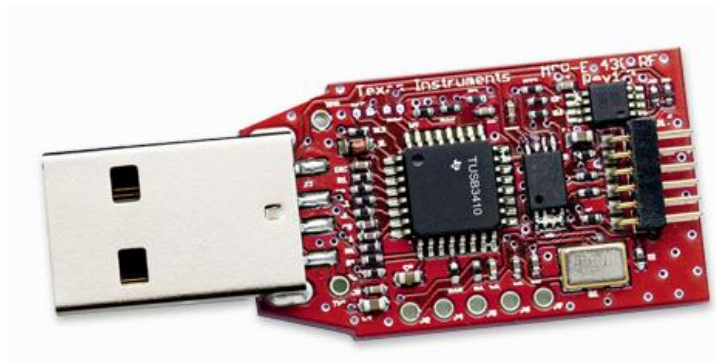


Figura 3: Grabador USB

El kit de baterías funciona con dos pilas AA y suministran una tensión de 3.6 V. En nuestro caso, como veremos más adelante, tendremos que realizar una modificación a este para poder realizar el sistema. El kit se puede ver en la Figura 4.



Figura 4: Kit de baterías

#### 4.1.2 Placa PCB

La placa PCB o placa de circuito impreso es una placa sobre la que se montan todos los componentes y se evita así el engorroso uso de cables. Esta placa ya tiene un diseño realizado con el software de libre distribución *FREPCB*. El archivo del diseño, llamado PCB.frb, contiene todo lo necesario para realizar la placa (en el caso presente se llevó al laboratorio de electrónica de la universidad para que realizaran la placa). Es posible que debido a la superficie que se quiera monitorizar la placa necesite alguna modificación. Si se diera el caso, el diagrama de conexiones de esta placa se puede ver en la Figura 5.

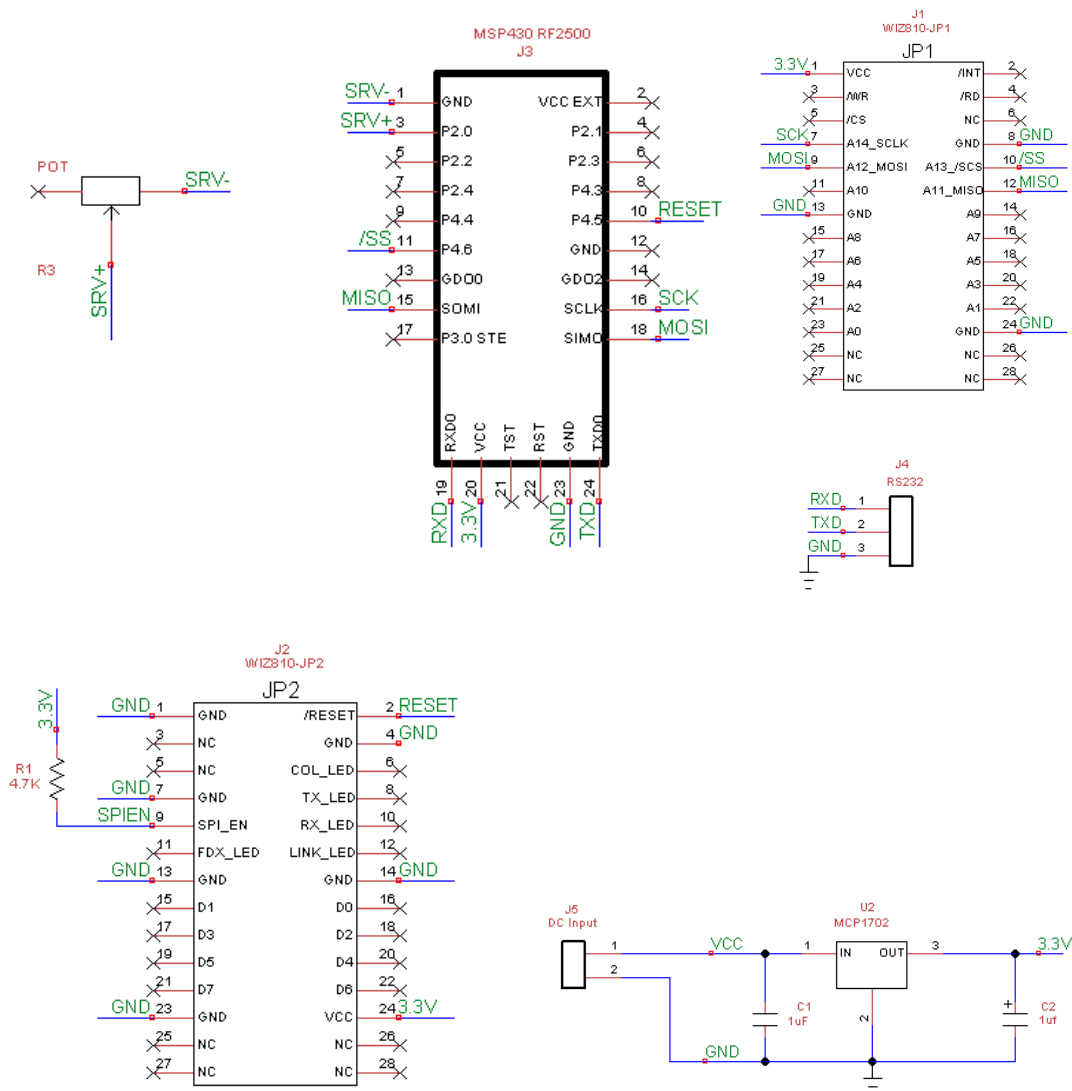


Figura 5: Esquema de conexiones del sistema



En caso de utilizar la placa ya desarrollada, el resultado debería ser parecido al de la Figura 6.

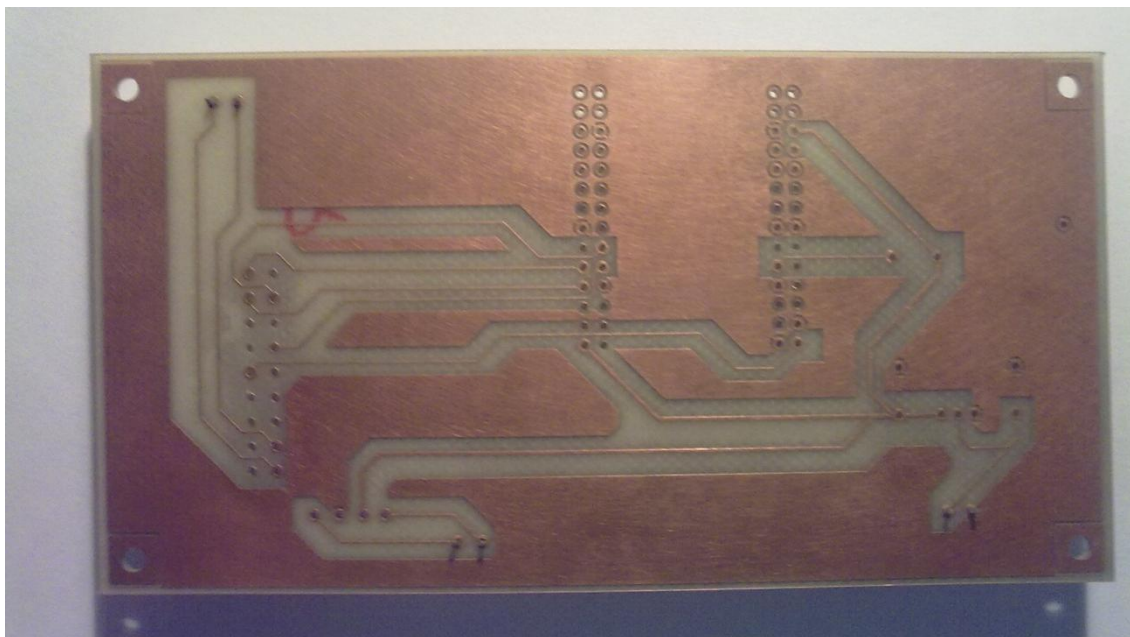


Figura 6: Imagen de la parte inferior de la PCB

### 4.1.3 Montaje del hardware

Para el montaje del hardware lo principal es tener un soldador de hierro para soldar los componentes a la placa, y estaño para realizar las conexiones entre la placa y el componente. Se recomienda tener todos los componentes antes de empezar el montaje del sistema. Será necesario también preparar el servo que mueve la superficie ya que será nuestro sensor.

#### 4.1.3.1 Soldar cables al potenciómetro

El primer paso a realizar será soldar dos cables unifilares a las conexiones del servo que se encuentran en la placa del servo. Será por tanto necesario tener bien localizadas las soldaduras del potenciómetro a la placa del servo antes de empezar a soldar. También es importante utilizar cable de diferente color para diferenciar entre los dos terminales.

Se debe soldar un cable a la conexión central del potenciómetro, que será la que vaya al amplificador operacional del microcontrolador, y otro cable a la conexión de tierra del servo. El resultado debería ser parecido al de la Figura 7.



Figura 7: Cables soldados al potenciómetro

Una vez soldados, hay que cerrar el servo. Se recomienda intentar sacar los cables por la misma salida de los cables del servo, como en la Figura 8. Si no fuera posible, habría que hacer unas pequeñas perforaciones en la tapa del servo y tras pasar el cable sellarlo con silicona térmica.



Figura 8: Servo cerrado con los cables extraídos

Habr  tambi n que poner en el otro extremo de los cables un conector polar hembra para conectarlo a la placa PCB del sistema. Es muy importante que el conector de polaridad hembra coincida con el del macho que ir  en la PCB ya que de lo contrario se alterar n las polaridades y podr a haber un problema muy grave. Eso quiere decir que se tiene que poner especial cuidado a la hora de poner los conectores de forma que la salida de tensi n del potenc metro vaya a la toma de tensi n del microcontrolador, y la toma de tierra a la toma de tierra.



Figura 9: Conector de polaridad hembra del servo

#### 4.1.3.2 Conector macho-hembra a los pins del m dulo MSP430 RF2500

El m dulo MSP430 RF2500 tiene una serie de pins que sirven para las distintas conexiones que se pueden realizar. Para mayor comodidad, se le soldar n a esos pins un conector de dos filas de 9 de paso 2.54 mm. Quedar  un m dulo como el de la Figura 10.

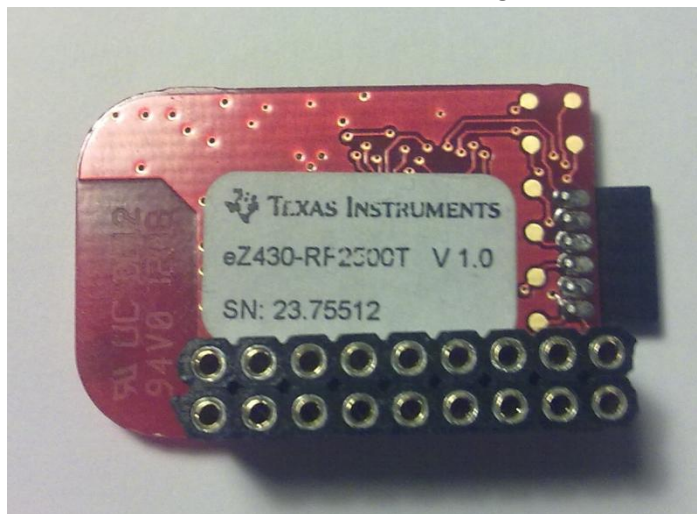


Figura 10: M dulo MSP430 RF2500 con el conector soldado

#### 4.1.3.3 Componentes soldados a la placa

El sistema funciona con una serie de componentes que deben ser soldados a la placa PCB. Se detallará a continuación como y donde se deben soldar los componentes. Los componentes deben estar por la parte superior de la placa (la parte sin cobre) y soldado por la parte inferior (la parte con cobre, que se ha visto ya en la Figura 6).

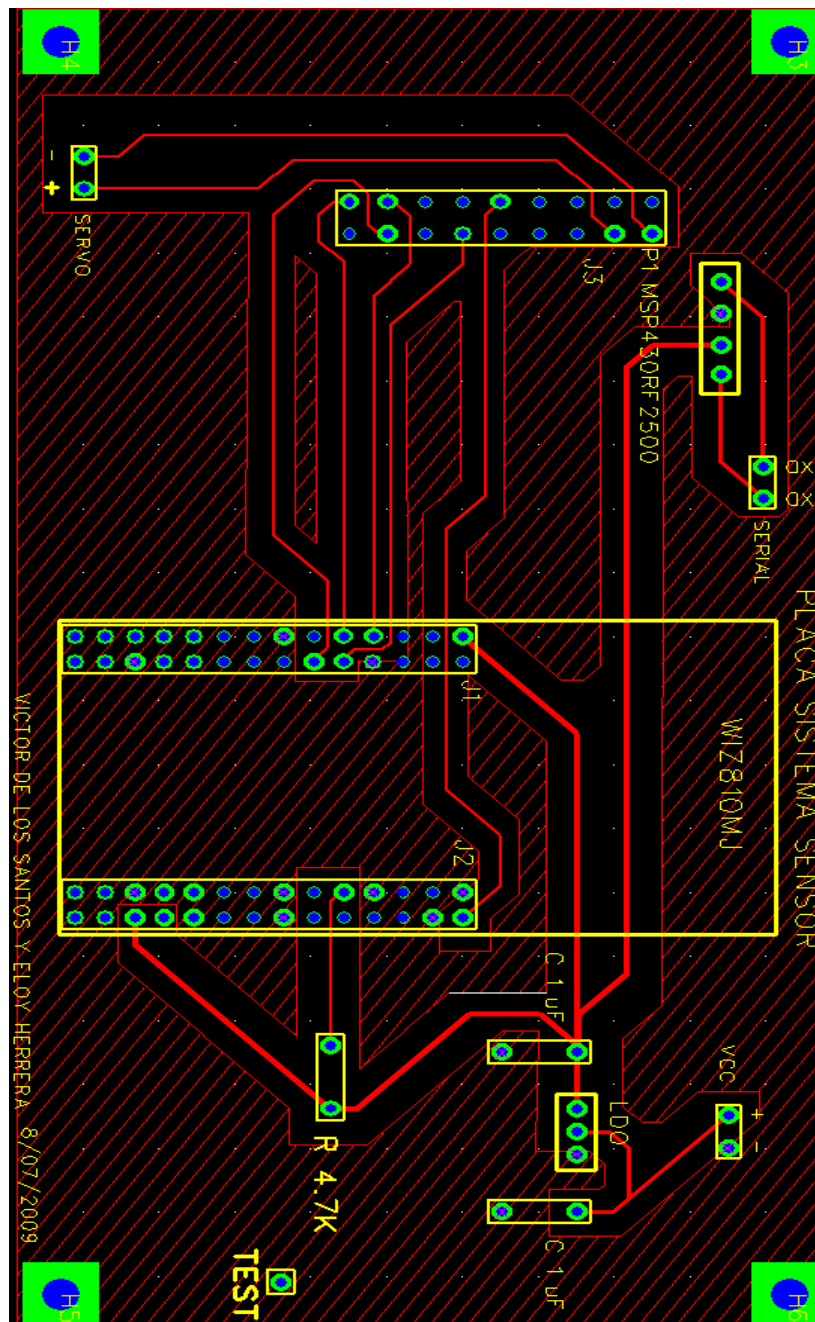


Figura 11: Conexiones de la PCB



Para saber exactamente donde va cada parte, en el esquema de la Figura 11 puede verse a que pieza corresponde cada agujero de la PCB.

Se empezará soldando los conectores con polaridad. Estos 3 conectores permitirán la entrada de la alimentación del sistema, la conexión serie de entrada y salida del microcontrolador y la conexión al potenciómetro. La Figura 12 muestra los conectores ya soldados a la placa.



Figura 12: Conectores de polaridad soldados a la PCB

A continuación se soldarán los condensadores cerámicos de 1  $\mu\text{F}$  (Figura 15), el regulador de tensión LDO (Figura 13) y la resistencia de 4.7  $\text{k}\Omega$  (Figura 14).

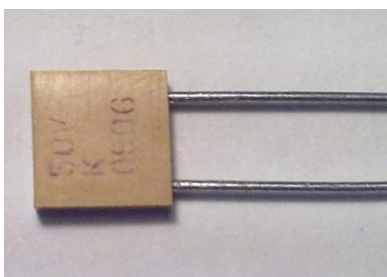


Figura 15: Condensador de 1  $\mu\text{F}$

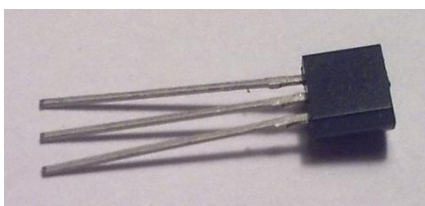


Figura 13: Regulador LDO de 3.3 V



Figura 14: Resistencia de 4.7  $\text{k}\Omega$

Se soldarán a la placa tal y como se ve en la Figura 16. Es muy importante que el regulador se suelde exactamente de la forma en que está, ya que las tres conexiones tienen una función diferente cada una y por tanto no pueden ir en cualquier conexión [4]. En los condensadores y la resistencia es indiferente la dirección en la que estén soldados.

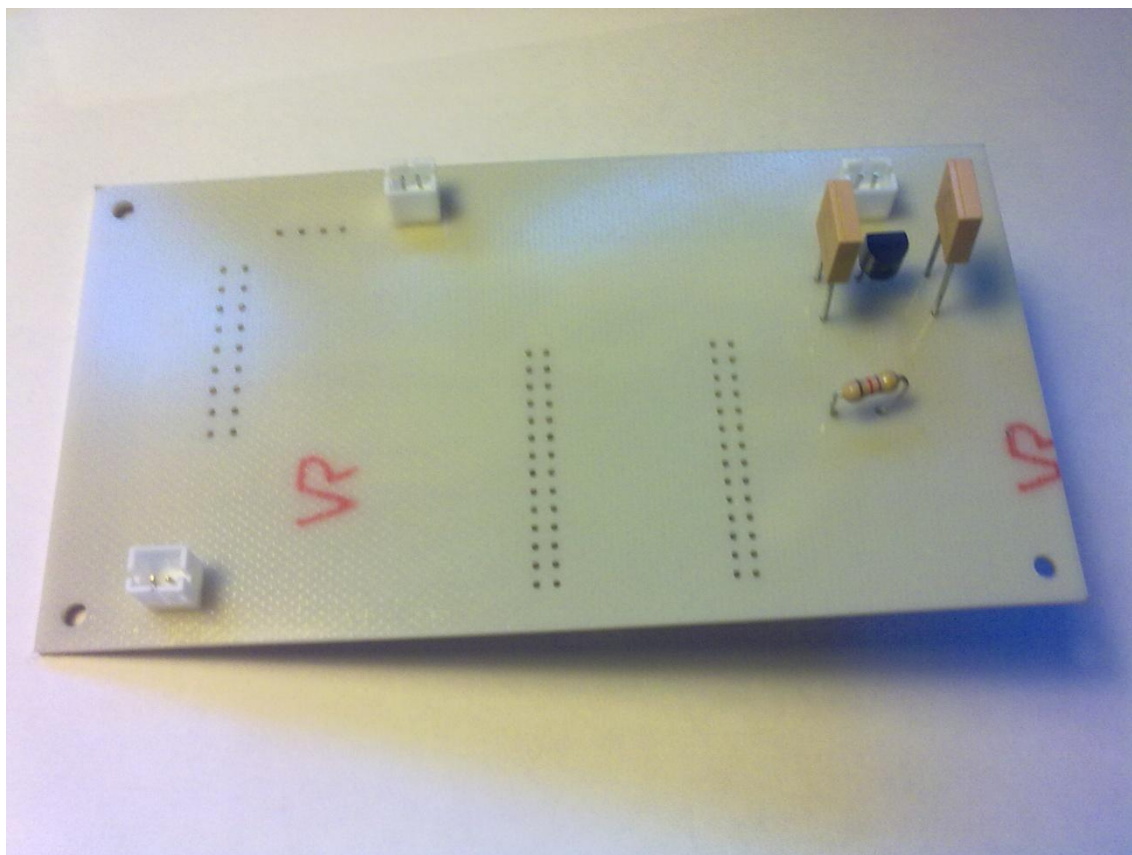


Figura 16: Regulador, condensadores y resistencias soldados a la placa

A continuación se soldarán los conectores macho-hembra de paso 2 mm y los macho-macho de paso 2.54 mm. Esto se realiza para no soldar ni el microcontrolador ni el módulo de Ethernet directamente sobre la placa, para no dañarlos y poder conectarlos como si de un zócalo se tratase. Además, aunque no tanto como la soldadura, aportan un gran agarre con los sistemas y por tanto no suponen un problema por desconexión. En los agujeros centrales van los conectores macho-hembra de paso 2 mm, que es donde irá el módulo ETHERNET; y en los del extremo los conectores macho-macho de paso 2.54 mm, donde se conectará el módulo MSP430 RF2500. El resultado se puede ver en la Figura 17.

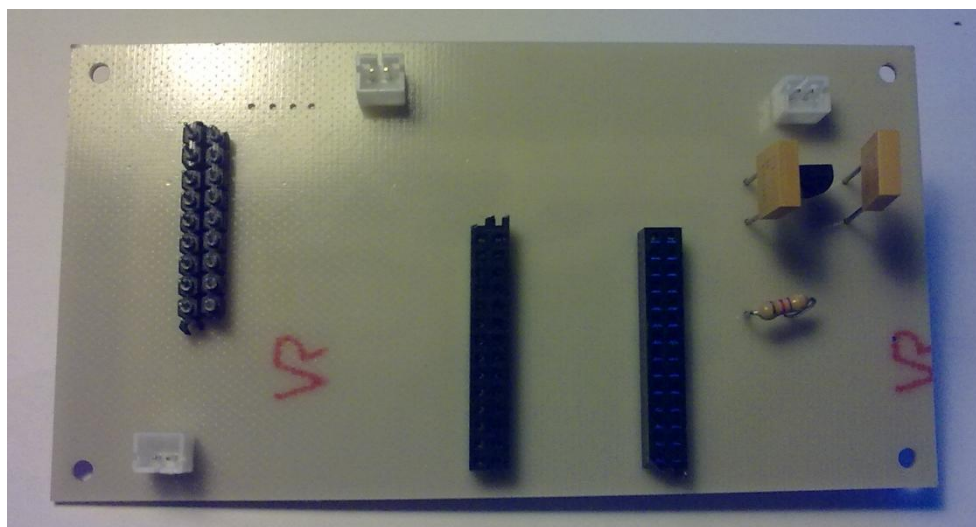


Figura 17: Placa con los pasos soldados

De esta forma, ya se pueden encajar los módulos MSP430 RF2500 y el módulo Ethernet WIZ810MJ, como se aprecia en la Figura 18.

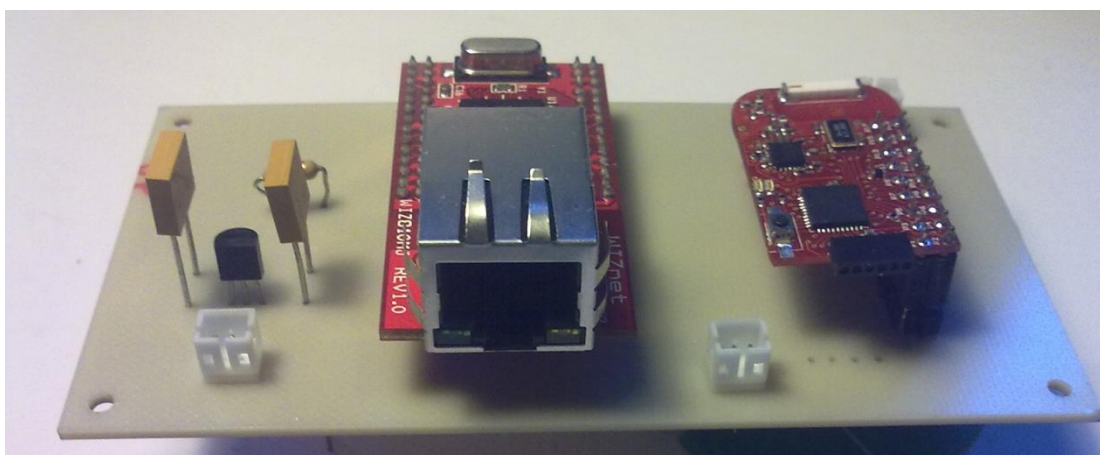


Figura 18: Módulos instalado sobre la placa

Finalmente, es necesario realizar una modificación al modulo de baterías. Debido a que el tipo de conector de alimentación y conexión de entrada y salida del puerto serie del microcontrolador es muy extraño, utilizaremos la placa que incorpora el sistema de baterías que tiene esa conexión. Para ello, tendremos que desoldar la placa por los puntos indicados en la Figura 19 (unos alicates para comprimir el punto de soldadura ayuda bastante).





El resultado debe ser algo similar a la imagen de la Figura 21.

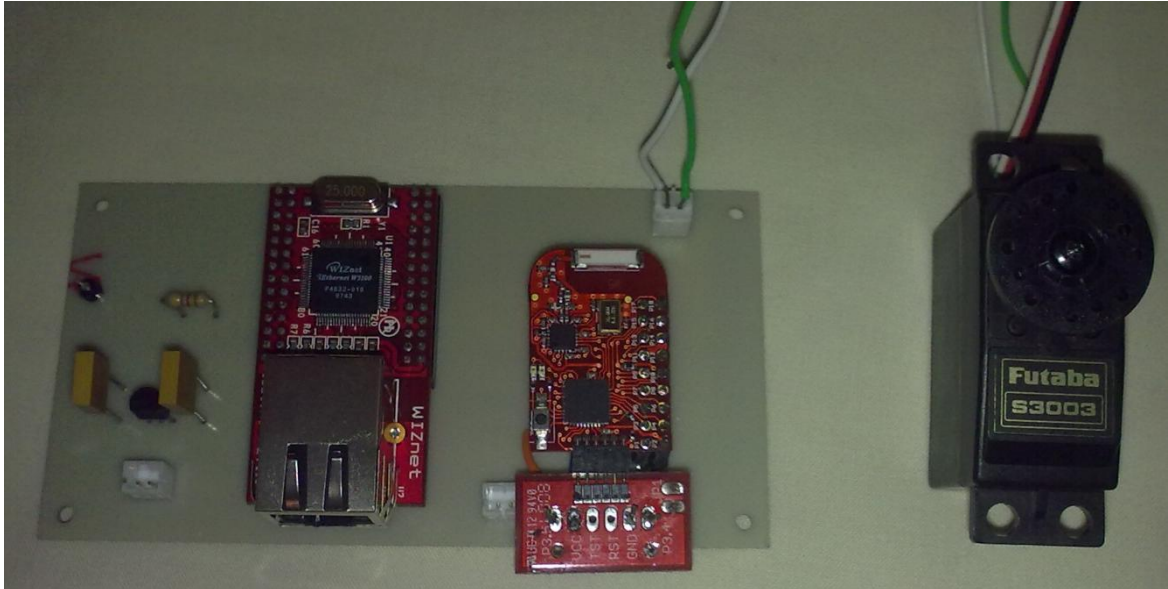


Figura 21: Placa con todos los componentes soldados y el Servo conectado

## 4.2 Preparación del Software

El software es el encargado de hacer que todo el hardware funcione. Es el que ordena que los diferentes componentes realicen sus funciones correctamente. En la Figura 22 se puede ver un esquema de las diferentes partes del software.

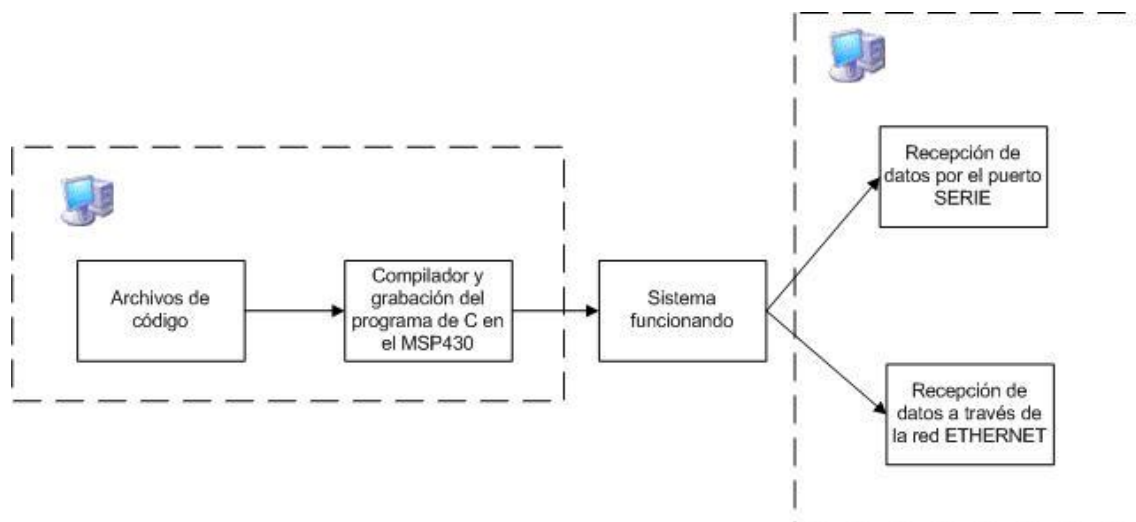


Figura 22: Diagrama de preparación del software para el funcionamiento del sistema

Para la implementación del software se necesitará:

- Archivos de código:
  - Código.c
  - W5100reg.h
  - Sistema.h
  - Ethernet.h
- Compilador de C compatible con microcontroladores MSP430
  - Recomendado: IAR Embedded Workbench Kickstart Edition v4.20 [1]
- Programa para recibir los datos por conexión SERIE (UART)
  - Recomendado: HyperTerminal Private Edition [2]
  - Labview
- Programa para recibir los datos por conexión ETHERNET (UDP)
  - Recomendado: HyperTerminal

En este manual se detallará el procedimiento a seguir con los archivos de código y se realizará una explicación de cómo utilizar el sistema. Sin embargo, aunque se recomienda la utilización de los programas aquí mencionados y con los cuales se explicará cómo utilizar el sistema, es posible utilizar otros programas que realicen funciones similares.

#### 4.2.1 Archivos de código:

El sistema funciona con los archivos Código.c, w5100reg.h, sistema.h y Ethernet.h. Su relación la vemos en la Figura 23.

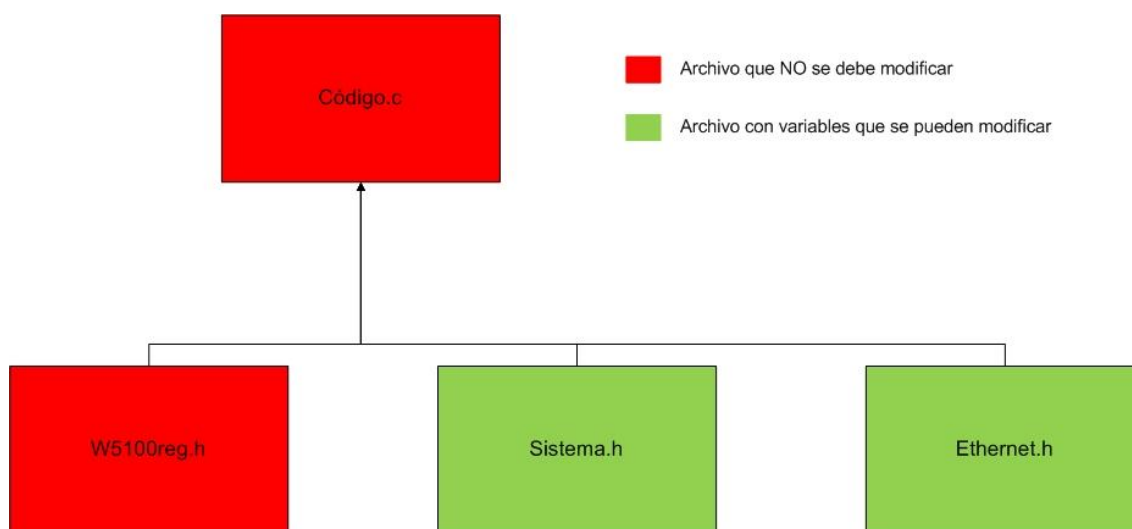


Figura 23: Diagrama de distribución de los archivos

El archivo código.c contiene la base del programa, las funciones y todos los procesos que controlan el sistema. No hay que modificar nada en este archivo si se quiere asegurar el correcto funcionamiento del sistema. Cualquier cambio en este no está contemplado y por tanto solo debe tocarse si se conoce con seguridad lo que se está cambiando en este.

El archivo w5100reg.h contiene las direcciones de los registros de configuración con los que funciona el módulo WIZ810MJ con el chip de Ethernet W5100 y los registros de control del SOCKET 0. No es necesario cambiar nada en este archivo. Si se deseara utilizar alguno de los otros 3 SOCKETS restantes se podrían incluir en este archivo los registros de control de esos SOCKETS (NO ES NECESARIO PARA EL FUNCIONAMIENTO DEL SISTEMA).

Los archivos sistema.h y Ethernet.h son los archivos que contienen las variables que se deben modificar para adaptar el sistema a otro sistema externo. Es importante remarcar que los datos tienen que ser números enteros, ya que el microcontrolador no está pensado para trabajar con decimales.

#### 4.2.1.1 Sistema.h

<b>Sistema.h</b>
- Posición mínima
- Posición máxima
- Tensión mínima (digital)
- Tensión máxima (digital)
- Identificador

Tabla 5: Variables del archivo Sistema.h

El archivo sistema.h consta de 6 constantes que definen al sistema que se quiere medir y que deben ser medidas y conocidas para que el sistema sensor funcione. La Tabla 5 muestra la lista de las variables.

##### 4.2.1.1.1 Variable “posmin” (Posición mínima)

Define la posición mínima de la superficie que se quiere medir. Debido a las características del microcontrolador y el desarrollo del programa, la posición mínima debe ser mayor o igual a cero, no pudiendo ser negativa. La posición se representará como un ángulo en grados.

##### 4.2.1.1.2 Variable “posmax” (Posición máxima)

Define la posición máxima de la superficie que se quiere medir. La posición se representará como un ángulo en grados.

##### 4.2.1.1.3 Variable “nmin” (Tensión mínima tras la conversión A/D) y variable “nmax” (Tensión máxima tras la conversión A/D)

Define la tensión mínima y máxima que sale del potenciómetro. Para obtener estas variables leer el **manual de calibración**.

##### 4.2.1.1.4 Variable “ident” (Identificador)

Define el identificador de la superficie que se está midiendo. Se puede asignar cualquier número, y este número aparecerá en el mensaje enviado con la información de la posición de la superficie. Debido a la programación, el rango de números posibles para el identificador estará entre 0 y 99 (siendo posible cambiarlo mediante la modificación del archivo código.c).

#### 4.2.1.2 Ethernet.h

El archivo Ethernet.h contiene todas las constantes que definen la configuración Ethernet del sistema, incluyendo la dirección IP del sistema, la puerta de enlace, máscara de subred,, dirección física (MAC), puerto del socket, dirección IP a la que se envían los datos por UDP y el puerto al que se envían. La localización de estas variables en la red se ve en la Figura 24.

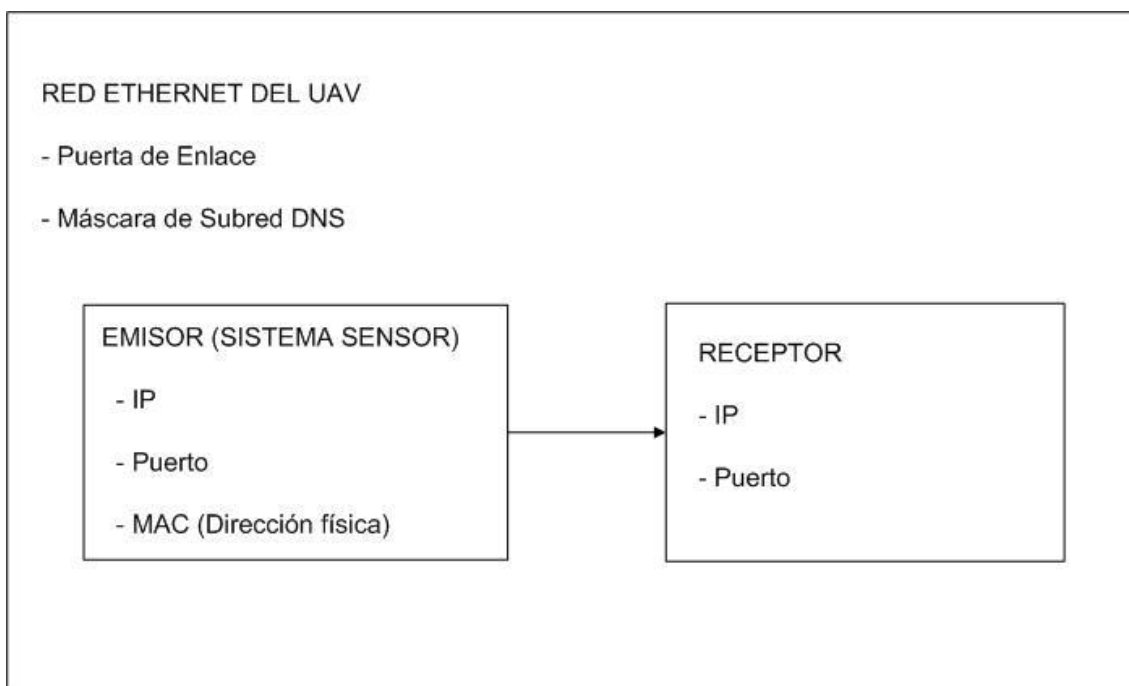


Figura 24: Diagrama de la red ETHERNET

##### 4.2.1.2.1 Puerta de enlace del sistema (GIP0.GIP.GIP2.GIP3)

Puerta de enlace del sistema ethernet en el que se integra el sistema sensor. Si tenemos una puerta de enlace del tipo 192.168.0.1, para cada parte se define una variable, donde:

- GIP0 = 192
- GIP1 = 168
- GIP2 = 0
- GIP3 = 1

#### 4.2.1.2.2 Máscara de subred DNS (DNS0.DNS1.DNS2.DNS3)

Máscara de subred DNS del sistema Ethernet en el que se integra el sistema sensor. Si tenemos una máscara de subred DNS del tipo 255.255.255.0, para cada parte se define una variable, donde:

- DNS0 = 255
- DNS1 = 255
- DNS2 = 255
- DNS3 = 0

#### 4.2.1.2.3 Dirección MAC (MAC0.MAC1.MAC2.MAC3.MAC4.MAC5)

Dirección MAC del sistema sensor. La MAC la define el usuario a la hora de implementar el sistema sensor. Cada parte de la dirección MAC se asigna con un número en formato hexadecimal. Para ello se pondrá el número en formato hexadecimal precedido de "0x". Si la dirección MAC es AA.11.22.33.44.55, cada parte se define como:

- MAC0 = 0xAA;
- MAC1 = 0x11;
- MAC2 = 0x22;
- MAC3 = 0x33;
- MAC4 = 0x44;
- MAC5 = 0x55;

#### 4.2.1.2.4 Dirección IP (IP0.IP1.IP2.IP3)

Dirección IP asignada al sistema sensor. Si tenemos una IP del tipo 192.168.0.5, para cada parte se define una variable, donde:

- IP0 = 192
- IP1 = 168
- IP2 = 0
- IP3 = 5

#### 4.2.1.2.5 Puerto del socket abierto (iport0 e iport1)

Las variables iport0 e iport1 definen el puerto abierto del socket udp por el que circula la información. Los puertos se definen como un número de 16 bits. Para fijar el puerto, será necesario pasar el puerto de formato decimal a formato hexadecimal. Tras esto, quedará un valor en formato hexadecimal formado por 4 dígitos, donde los dos primeros serán la variable iport0 y los dos últimos serán la variable iport1. Se guardarán en su respectiva variable precedidos por "0x". Por ejemplo, si se quiere utilizar el puerto 5000, en hexadecimal es el puerto 1388. Por tanto, quedará de la siguiente forma:

- iport0 = 0x13
- iport1 = 0x88

#### 4.2.1.2.6 Dirección IP a la que se envía la información (IPD0.IPD1.IPD2.IPD3)

Dirección IP a la que se quiere emitir la información mediante el protocolo UDP. Si tenemos una IP del tipo 255.255.255.255 (IP para emitir en BROADCAST, es decir, por toda la red) para cada parte se define una variable, donde:

- IPD0 = 255
- IPD1 = 255
- IPD2 = 255
- IPD3 = 255

Por supuesto, es posible emitir a una ip en concreto y no realizar un broadcast, simplemente cambiando esta dirección IP.

#### 4.2.1.2.7 Puerto por el que se recibe la información (dport0 e dport1)

Las variables dport0 e dport1 definen el puerto abierto del socket udp al que envía la información. Los puertos se definen como un número de 16 bits. Para fijar el puerto, será necesario pasar el puerto de formato decimal a formato hexadecimal. Tras esto, quedará un valor en formato hexadecimal formado por 4 dígitos, donde los dos primeros serán la variable dport0 y los dos últimos serán la variable dport1. Se guardarán en su respectiva variable precedidos por "0x". Por ejemplo, si se quiere utilizar el puerto 4369, en hexadecimal es el puerto 1111. Por tanto, quedará de la siguiente forma:

- dport0 = 0x11
- dport1 = 0x11

#### 4.2.2 Compilación y grabación del código en el microcontrolador MSP430

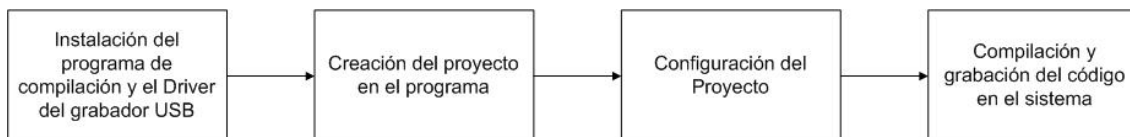


Figura 25: Diagrama de instalación y preparación del compilador y grabador

Durante este capítulo se explicará y detallará la compilación y grabación del programa de funcionamiento en el módulo MSP430 RF2500. La Figura 25 muestra un diagrama del proceso que se seguirá. Para ello, se utilizará el programa IAR Embedded Workbench Kickstart Edition v4.2.

##### 4.2.2.1 Instalación del programa

Este programa requiere una instalación instalar directamente desde el CD que viene con el kit de desarrollo MSP430 RF2500 o descargado de la página web de Texas Instruments. Para instalarlo solo es necesario ejecutar el archivo de instalación y seguir los pasos que indica el programa.

##### 4.2.2.2 Instalación del driver

Una vez instalado, será necesario introducir el módulo USB que conecta los módulos del MSP430 con el ordenador para poder grabarlos, y que tal como está descrito en el manual de hardware, viene incluido en el kit. Al insertarlo, se nos preguntará la instalación del driver. Para ello, si no lo ha detectado automáticamente, será necesario buscarlo manualmente. El driver se encuentra en la siguiente ruta (si se ha dejado la que viene en la instalación por defecto):

C:\Archivos de programa\IAR Systems\Embedded Workbench 5.0\430\drivers\TIUSBFET\WinXP

##### 4.2.2.3 Compilación y grabación del proyecto

Tras la instalación del driver, abriremos el programa (icono del programa en la Figura 26).



Figura 26: Icono del programa

Al abrirlo, nos aparecerá una pantalla con diversas opciones, donde se hará click en la primera opción (*Create new Project In Current Workspace*), como indica la Figura 27.



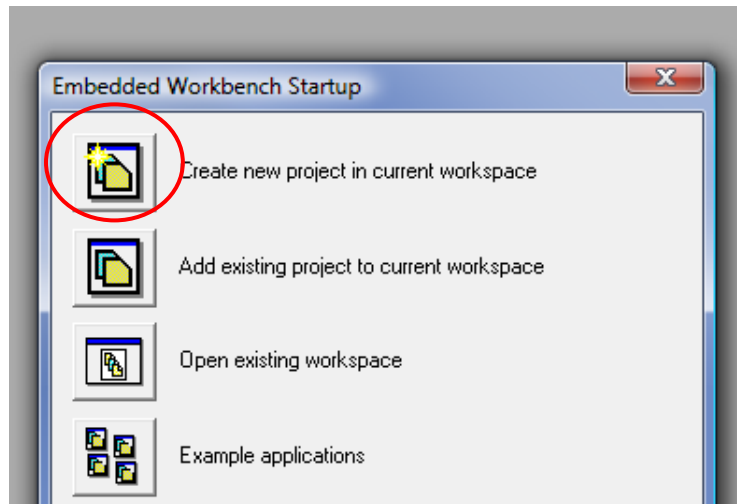


Figura 27: Ventana de inicialización del IAR Embedded Workbench

Tras esto aparecerá una pantalla Figura 28 donde seleccionaremos si no están seleccionadas, las siguientes opciones:

- **Tool Chain: MSP430**
- **Empty Project**

Y haremos click en **OK**.

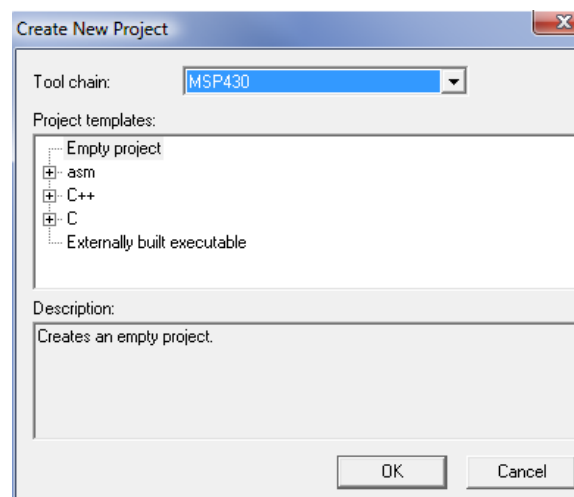


Figura 28: Creación de nuevo proyecto del IAR Embedded Workbench

Nos aparecerá una ventana donde guardar el nuevo proyecto. Seleccionaremos la ubicación deseada y le daremos el nombre que queramos. Tras esto, haremos click en guardar. Aparecerá la pantalla del programa vacía. Ahora se tienen que insertar los archivos del programa. Se recomienda guardarlos en la misma carpeta donde esté el proyecto que se ha creado con el programa. Para ello haremos click en la ventana **Project** y a continuación en **Add Files** (ver Figura 29).

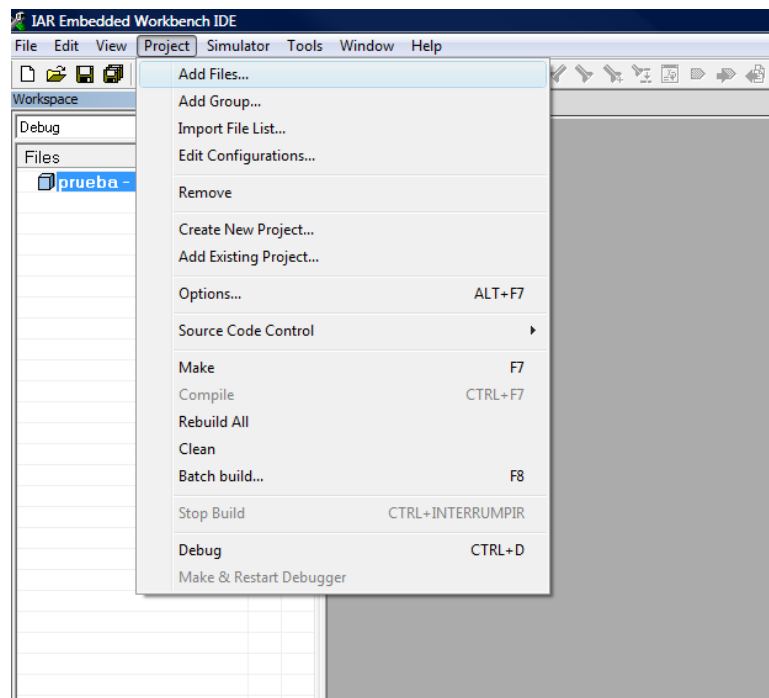


Figura 29: Añadir ficheros al proyecto

Añadiremos los cuatro archivos de código (Código.c, W5100reg.h, Sistema.h y Ethernet.h). Ahora debería haber una imagen como la Figura 30, donde se ven ya incluidos los 4 ficheros.

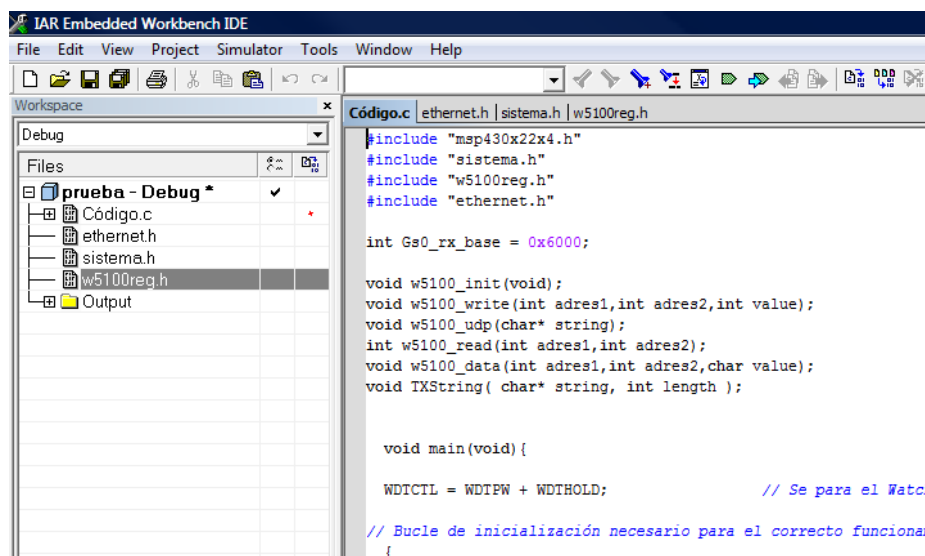


Figura 30: Proyecto con los ficheros de código añadidos

Ahora hay que configurar el proyecto para que guarde el código correctamente en el módulo MSP430. Para ello, haremos click en **Project** y a continuación en **Options** (Figura 31). A continuación cambiamos solo los parámetros que aquí se indican, y el resto se deja tal y como está.

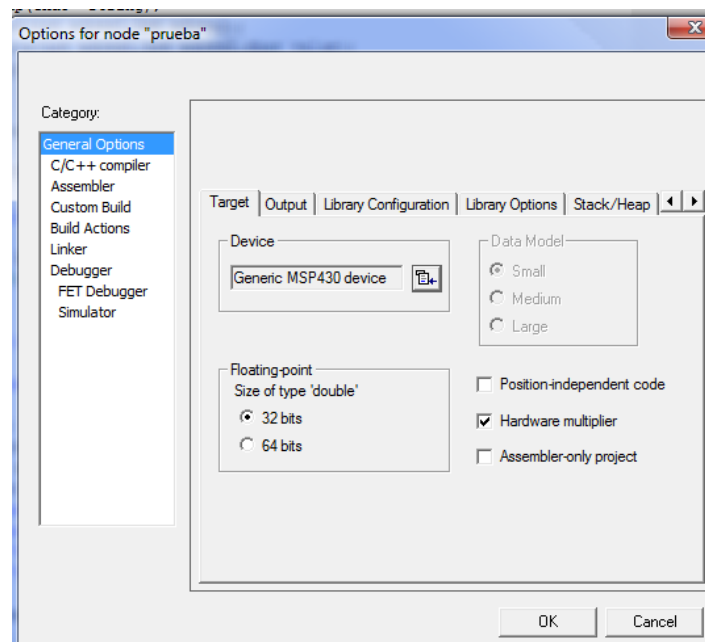


Figura 31: Opciones de proyecto

Categoría **General Options:**

- Target
  - o Device: **MSP430F2274**
- Library Configuration
  - o Library: **CLIB**

Categoría **C/C++ Compiler:**

- Output
  - o Module Type: **Override Default**
- List
  - o **Output File** Marcado
    - **Assembler Mnemonics** Marcado

Categoría **Debugger**

- Setup
  - o Driver: **FET DEBUGGER**

Tras esto, le damos a OK y el proyecto ya estará configurado. Solo queda grabar el programa en el MSP430. Para ello tiene que estar el módulo MSP430 RF2500 insertado correctamente en el grabador USB (Figura 32).

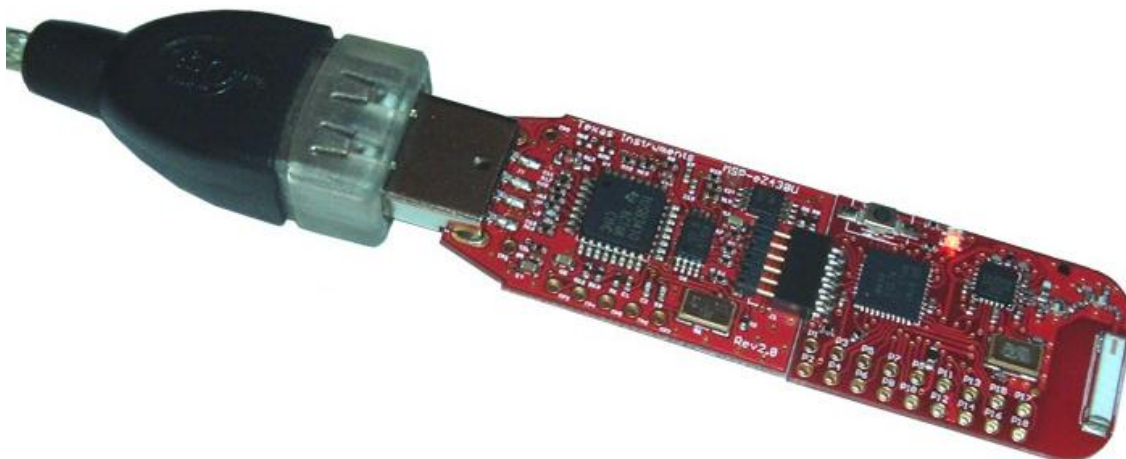


Figura 32: Grabador USB conectado al módulo MSP430 RF2500 y a un conector USB del Ordenador

Antes será necesario guardar el Workspace. Para ello iremos a **File** y a continuación **Save Workspace**. Seleccionaremos donde queremos guardar el Workspace (recomendamos usar la misma carpeta del proyecto). Cuando le hayamos dado un nombre, ponemos guardar.

Ahora se procederá a guardar el código en el módulo del MSP430. Para ello, iremos a **Project** y a **Debug** (también podemos hacerlo más rápido haciendo la combinación de teclas **CONTROL+D**).

Ahora, y si se han seguido todos los pasos correctamente, aparecerá la pantalla de debugación. Eso quiere decir que el programa ya está cargado en el microcontrolador. Para parar la debugación hacemos click en el icono marcado en la Figura 33 o utilizamos la combinación de teclas **CONTROL+SHIFT+D**.

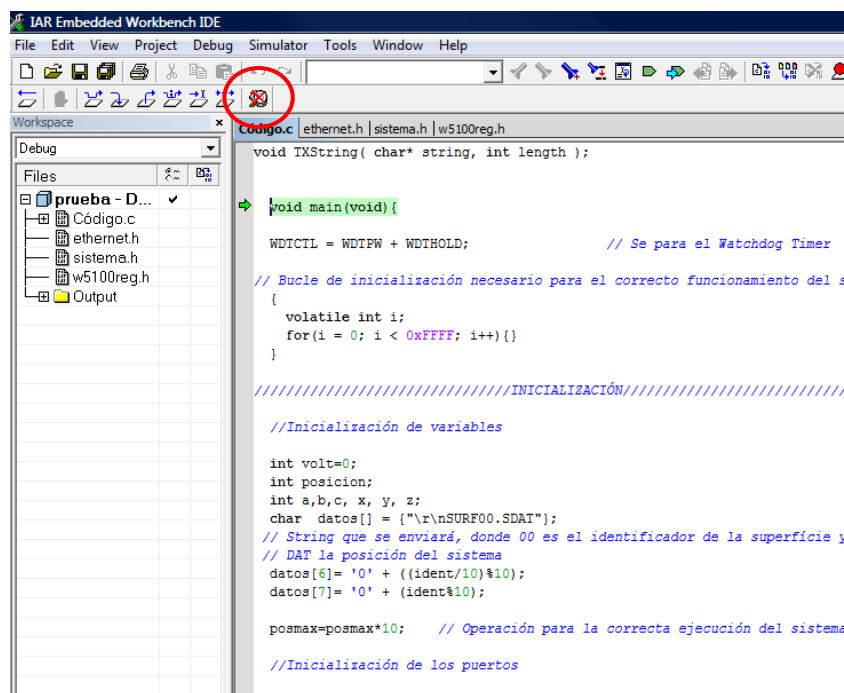


Figura 33: Pantalla de Debugación e Icono de parada de debugación

El programa ya está guardado en el microcontrolador y se encuentra totalmente funcional.

## 5 MANUAL DE CALIBRACIÓN

El manual de calibración permite ajustar nuestro sistema de monitorización a la superficie a controlar. La función principal de este manual es la obtención de las variables “nmax” y “nmin” que el sistema necesita para su correcto funcionamiento. Existen dos formas de obtener las variables y se exponen ambas, siendo la más recomendada por su precisión la **OPCIÓN 2**.

### 5.1 Opción 1:

Estas tensiones se calcularán con las siguientes ecuaciones:

$$nmin = 1023 * \frac{V_{min} - V_{refmin}}{V_{refmax} - V_{refmin}}$$

Donde  $V_{refmin}$  y  $V_{refmax}$  están definidas en la programación del conversor ADC y en este caso valen  $V_{refmin} = 0 \text{ V}$  y  $V_{refmax} = 2500 \text{ mV}$ ; y  $V_{min}$  es la tensión entre la salida central y la toma de tierra del potenciómetro cuando la tensión en el potenciómetro del servo es mínima (que será cuando la posición del potenciómetro sea máxima). Para medir la tensión del potenciómetro se utilizará un multímetro o un voltímetro cualquiera.  $[V_{min}] = \text{mV}$ .

$$nmin = 1023 * \frac{V_{max} - V_{refmin}}{V_{refmax} - V_{refmin}}$$

Donde  $V_{refmin}$  y  $V_{refmax}$  están definidas en la programación del conversor ADC y en este caso valen  $V_{refmin} = 0 \text{ V}$  y  $V_{refmax} = 2500 \text{ mV}$ ; y  $V_{max}$  es la tensión entre la salida central y la toma de tierra del potenciómetro cuando la tensión en el potenciómetro del servo es máxima (que será cuando la posición del potenciómetro sea máxima). Para medir la tensión del potenciómetro se utilizará un multímetro o un voltímetro cualquiera.  $[V_{max}] = \text{mV}$ .

### 5.2 Opción 2:

En esta opción, aunque se necesita utilizar componentes de hardware y software, no se hará una diferenciación entre la preparación del hardware y el software ya que es un proceso que se puede realizar fácilmente a la misma vez.

El kit de desarrollo MSP430 RF2500 contiene dos módulos con el microcontrolador MSP430F2274. Para el desarrollo del sistema prototipo, solo se utiliza uno de los módulos. El otro módulo está configurado para poder obtener la tensión mínima y máxima tras la conversión A/D y emitir esa información por serie. Se puede conectar el potenciómetro del servo a ese módulo que ya está preparado para ello y obtener así las variables necesarias. La Figura 34 muestra un diagrama de conexiones para la obtención de las variables.

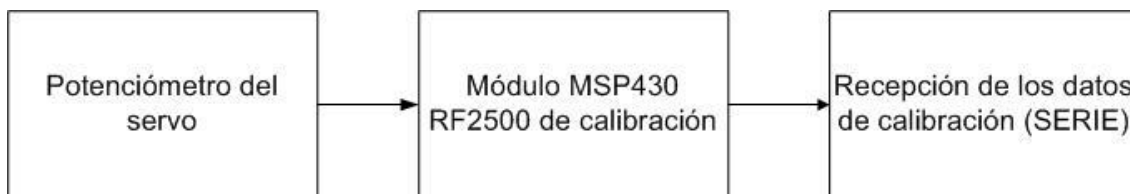


Figura 34: Diagrama de obtención de las tensiones máximas y mínimas

Teniendo en cuenta que el servo ya tiene los dos cables soldados al potenciómetro, antes de conectarlos a la placa definitiva los conectamos al módulo MSP430 RF2500 de calibración. Para ello, conectamos el cable de tensión del potenciómetro al PIN 3 del módulo MSP430 RF2500 de calibración, y el cable GND al PIN 1. Para facilitar la conexión, el módulo de calibración tiene soldado un paso al que conectar los cables. En la Figura 35 se ven los pins donde conectar el potenciómetro.

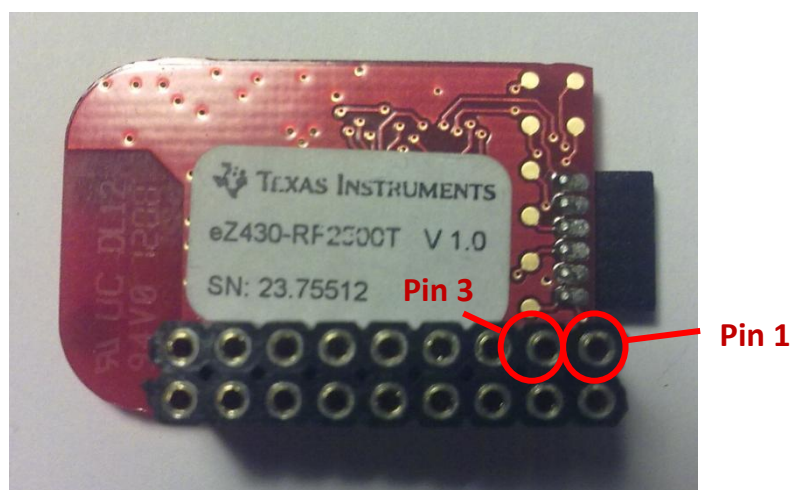


Figura 35: Módulo MSP430 RF2500 de calibración

Tras la conexión del potenciómetro, hay que recibir los datos mediante serie. El módulo de calibración se puede conectar fácilmente al grabador de USB que viene disponible con el kit de desarrollo y utilizar la conexión serie, tal como se ve en la Figura 36.

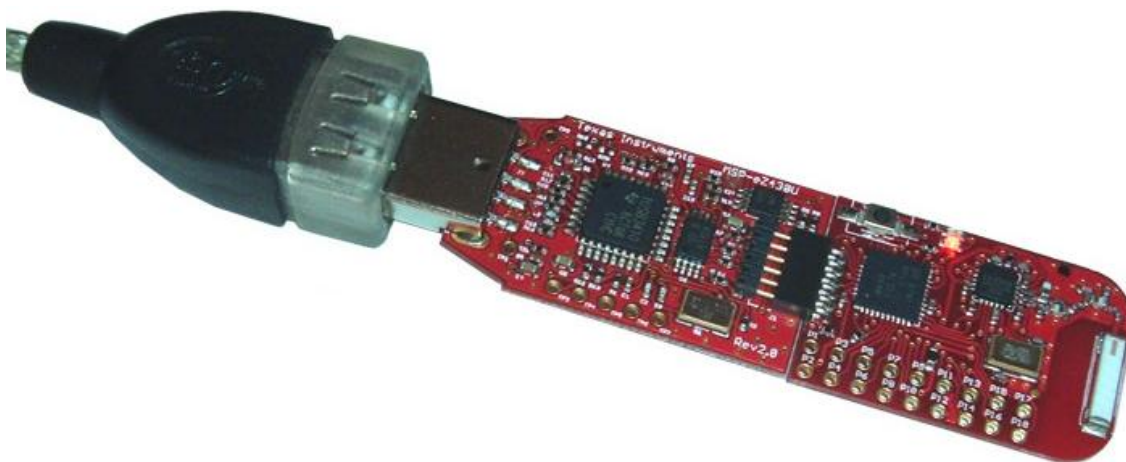


Figura 36: Grabador USB conectado al módulo MSP430 RF2500 y a un conector USB del Ordenador

Para configurar la conexión serie, se recomienda ver el manual de usuario, el apartado sobre la conexión SERIE.

Una vez realizada la conexión, se recibirán un mensaje de 11 bytes de la forma:

"VOLT.XXXX"

El mensaje contiene la información de la tensión tras su conversión a formato digital. Una vez establecida la conexión serie, se debe guardar toda la información recibida en un archivo de texto que contenga todos los valores del sistema. Para realizar eso en HyperTerminal, una vez realizada la conexión y estar recibiendo los datos, se debe ir a la ventana Transfer, y hacer click en capture file. Elegimos la ruta de destino y desde ese momento toda la información recibida se guardará en un fichero de texto. La Figura 37 muestra la opción de capturar texto.

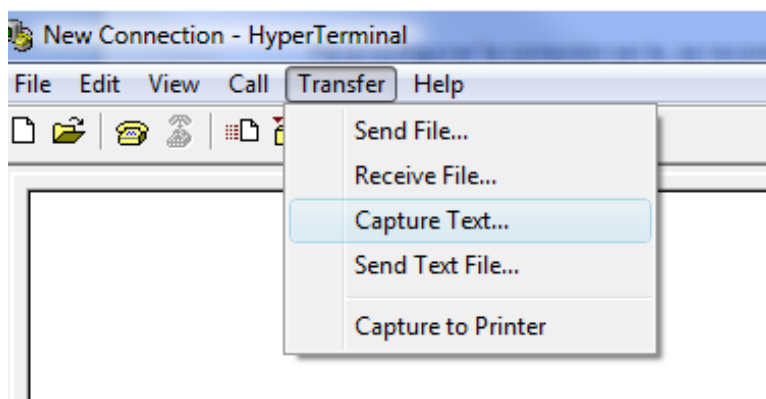


Figura 37: Capturar texto con HyperTerminal



Para obtener las variables, se colocará la superficie para la que se está adaptando el sistema en su posición máxima, dejándola durante unos segundos en esa posición. A continuación, se desplazará a la posición mínima, dejándolo también durante unos segundos.

Ahora será necesario buscar en el fichero de texto donde se han guardado los datos, los datos máximos y mínimos, y esas serán las variables “nmax” y “nmin”, siguiendo el diagrama de la Figura 38. Las variables “nmax” y “nmin” pueden dar unos valores en un rango entre 0 a 1023. Sin embargo, lo normal es que para los potenciómetros de los servos, la variable “nmin” da valores entre 50 y 100, y “nmax” entre 600 y 1000. Es importante saber que la tensión máxima que es capaz de procesar el sistema es de 2.5 V. En caso de que la tensión máxima que dé el potenciómetro del servo sea mayor a 2.5 V, será necesario realizar cambios en el código. Para más información mirar el documento DISEÑO DETALLADO, en el apartado de Diseño de software.

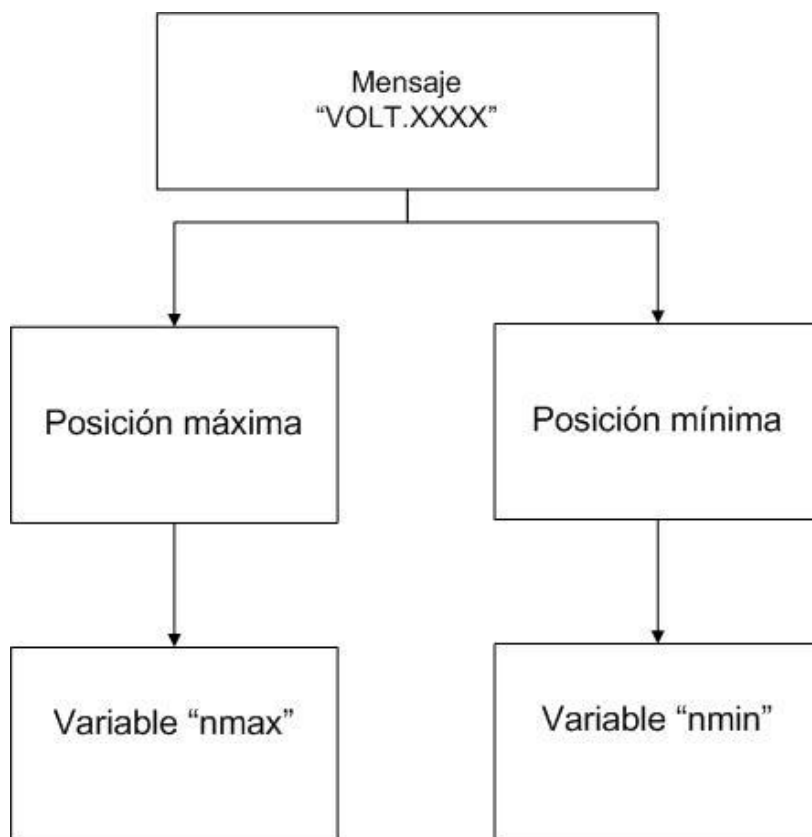


Figura 38: Diagrama de obtención de tensiones

Si se deseara, se puede compilar y grabar el código de calibración en un nuevo módulo, utilizando el archivo “Calibracion.c” para ello. Para compilar y grabar en el MSP430 revisar el capítulo 4.3 Compilación del código en el microcontrolador MSP430.

## 6 MANUAL DE USUARIO

Este manual detalla el funcionamiento normal del sistema y como obtener y utilizar los datos que este emite.

### 6.1 Mensaje transmitido

El sistema transmite constantemente un mensaje con la información de la superficie para la cual ha sido adaptada. El mensaje, un mensaje de 1 bytes, sigue un patrón como el siguiente

“SURFYY.±XXX”

Este mensaje contiene una cabecera, SURF, para identificar que se está transmitiendo la posición de una superficie. Le sigue un identificador numérico que marca a qué superficie se refiere el sistema, es decir, que superficie es de la que se está obteniendo la información. El número del identificador se lo dará el ingeniero que esté preparando el sistema, tal como está detallado en el manual de instalación. Tras el identificador, un punto y a continuación un símbolo que indica si la posición recibida es negativa o positiva (dependiendo de la posición en que se encuentre la superficie). Finalmente, llega el valor numérico que indica la posición de la superficie.

Por ejemplo, si la superficie es el alerón derecho, que tiene un identificador 03, y está en la posición 21º, el mensaje recibido será:

“SURF03.+021”

### 6.2 Recepción de datos por la conexión SERIE

La conexión SERIE se utilizará principalmente como una conexión de pruebas y para controlar el correcto funcionamiento del sistema. La Figura 39 representa el diagrama para la realización de una conexión serie.



Figura 39: Conexión por serie al sistema

Si el sistema está conectado por el puerto serie a algún ordenador, es muy sencillo comprobar los datos que se están recibiendo. Para ello se puede utilizar cualquier programa capaz de leer datos recibidos por un puerto SERIE, como Labview o HyperTerminal Private Edition (el programa HyperTerminal que viene incluido en todos los Windows excepto en Windows Vista también sirve y funciona exactamente igual que el aquí descrito).

Para configurar la conexión serie en cualquier programa, es necesario conocer los parámetros con los que está configurado el sistema. Aparecen en la Tabla 6:

Bits por Segundo	9600
Bits de Datos	8
Paridad	Ninguna
Bits de Parada	1

Tabla 6: Información de la conexión SERIE del sistema

### 6.2.1 EJEMPLO: HyperTerminal Private Edition

En este caso, se explicará cómo obtener los datos a través de la conexión SERIE mediante el programa **HyperTerminal Private Edition**. Este programa, de libre distribución, se puede descargar por internet [1]

Ejecutamos el archivo de instalación y seguimos los pasos para llevarla a cabo. Tras la instalación, ejecutamos el programa (icono de la Figura 40).



Figura 40: Icono del programa HyperTerminal Private Edition

Al abrirlo por primera vez, nos pedirá que introduzcamos datos del teléfono para configurar las conexiones. Dejamos libres todos los campos y damos a **OK** a todas las ventanas hasta que aparezca la pantalla de **New Connection – Connection Description** (Figura 41).

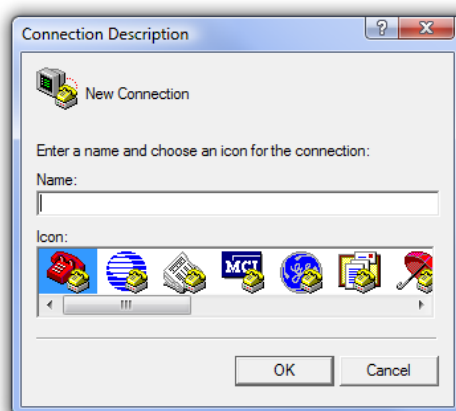


Figura 41: Pantalla de inicialización de la conexión

Le asignamos el nombre y el icono que deseemos y le damos a **OK**. Tras esto, aparece la ventana **Connect To**. Dejamos todos los campos libres y seleccionamos en la pestaña **Connect Using** la entrada COM a la que esté conectado el sistema.

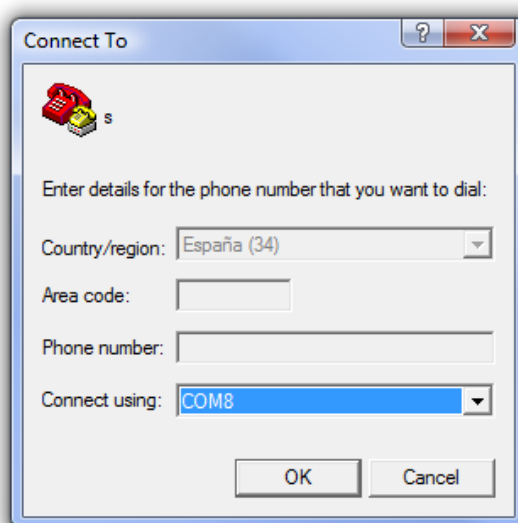


Figura 42: Selección del puerto de entrada de la conexión

En la siguiente ventana (Figura 43) podemos configurar la conexión SERIE. Siguiendo las características de la tabla 1, ponemos en cada casilla la configuración del sistema.

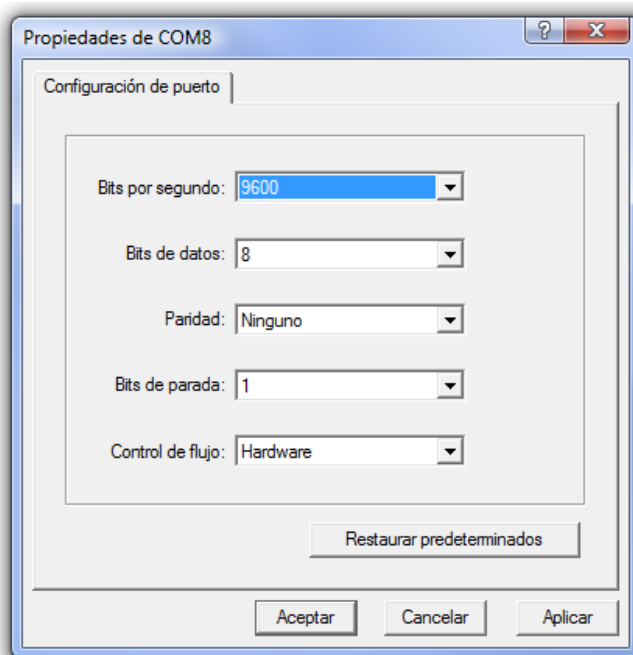


Figura 43: Configuración del puerto serie

Le damos a **Aceptar** y la conexión ya estará creada. Si el MSP430 está conectado y funcionando, empezaremos a recibir los datos, apareciendo continuamente como lo mostrado en la Figura 44. La conexión SERIE se ha realizado con éxito.

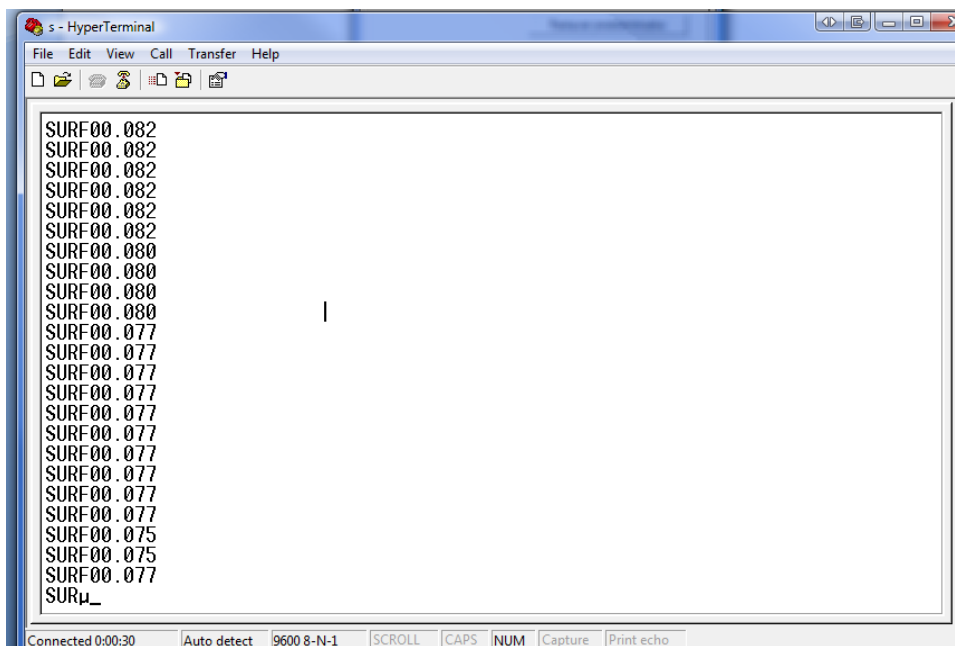


Figura 44: Recepción de datos del sistema sensor mediante la conexión SERIE

El mensaje empieza con un identificador de la superficie (SURF00) donde 00 es el número de la superficie medida. La parte de la derecha del punto es la que indica la posición de la superficie en grados.

Si se deseara, se puede guardar toda esta información con la opción Capture File, en el menú Transfer.

### 6.3 Recepción de datos por la conexión ETHERNET

Para la configuración de la conexión ETHERNET es necesario tener una red ETHERNET sobre la que incluir el sistema. Será necesario haber configurado entonces la red Ethernet mediante las diferentes variables que se han introducido en el archivo ETHERNET.h (ver Manual de Instalación para más información).

Será entonces necesario configurar la red Ethernet. Como cada una varía, no se va a detallar el proceso a seguir pues cada red es diferente. Solo es importante conocer que, por defecto, tiene la configuración de la Tabla 7.

Puerta de enlace	172.26.0.1
DNS	255.255.255.0
Dirección física MAC	AA.10.20.30.40.50
IP del sistema	172.26.0.111
Puerto abierto de acceso al sistema	5000
Protocolo	UDP
Socket Abierto	SOCKET 0
IP de destino	255.255.255.255 (BROADCAST)
Puerto de destino	4369

Tabla 7: Configuración del Ethernet del sistema por defecto

### 6.4 I.C.D

La Tabla 8 muestra el ICD del sistema

MENSAJE	"SURFXX.±YYY"
VELOCIDAD	9600 bits/segundo
Longitud del mensaje	11 bytes
Bits de datos	8 bits
Protocolo Ethernet	UDP broadcast
Puerto de destino	4369

Tabla 8: ICD



## 7 BIBLIOGRAFIA

[1] Hyper Terminal Private Edition:

<ftp://ftp.rdsor.ro/pub/Windows/Utils/HyperTerminal%20Private%20Edition%206.3/hpte63.exe>

[2] IAR Embedded Workbench Kickstart Edition,

<http://focus.ti.com/docs/toolsw/folders/print/iar-kickstart.html>

[4] *A Low Dropout Positive Voltage Regulator Product Brief*, Ref. MCP1702, MICROCHIP Technology Inc., 2005

[5] *W5100 Datasheet v. 1.1.8*, W5100\_Datasheet\_v1\_1\_8.pdf, WIZNET Inc., 4/03/2009

[6] *MSP430x22x2, MSP430x22x4 MIXED SIGNAL MICROCONTROLLER*, msp430f2274.pdf, Texas Instruments Inc. Mayo de 2009



## 8 GLOSARIO

**IP** = Internet Protocol

**DNS** = Domain Name System

**MAC** = Media Access Control address

**A/D** = Analógico / Digital

**UDP** = User Datagram Protocol

**PCB** = Printed Circuit Board

**UAV** = Unmanned Aerial Vehicle

**LDO** = Low Drop Out





Diseño de  
Sistema Sensor  
Para superficies  
de UAV

---

Ref.:  
Iss./Rev.: D01  
Date: 09/07/2009

[END OF DOCUMENT]

## ANEXO 5: Capacidad de Ampliación del Sistema

	Name	Signature	Date
Proyectista 1	Eloy Herrera		07/07/09
Proyectista 2	Víctor de los Santos		07/07/09
Director	Jorge Ramírez Alcantara		07/07/09

<b>1</b>	<b>INTRODUCCIÓN .....</b>	<b>5</b>
<b>2</b>	<b>OBJETIVO .....</b>	<b>6</b>
<b>3</b>	<b>CAPACIDAD DE AMPLIACIÓN .....</b>	<b>7</b>
3.1	Sistema de Alimentación .....	7
3.2	Conexión Serie .....	7
3.3	Conexión ETHERNET .....	8
3.4	Procesado de datos (MSP430 RF2500):.....	9
3.4.1	Funcionalidades de los pins libres.....	9
3.4.2	Hardware adicional .....	11
3.4.3	Características accesibles por SOFTWARE.....	11
<b>4</b>	<b>EJEMPLOS DE AMPLIACIÓN .....</b>	<b>12</b>
4.1	Monitorización de otras magnitudes .....	12
4.2	Monitorización de la señal recibida por el servo.....	12
4.3	Conexión a un POD subalar .....	12
4.4	Monitorización del voltaje.....	13
<b>5</b>	<b>BIBLIOGRAFÍA .....</b>	<b>14</b>
<b>6</b>	<b>GLOSARIO .....</b>	<b>15</b>

## *Índice de Tablas*

TABLA 1: CONSUMO DE CORRIENTE UTILIZADO Y DISPONIBLE .....	7
TABLA 2: CARACTERÍSTICAS DE LA CONEXIÓN SERIE DEL MÓDULO MSP430 RF2500 .....	8

## *Índice de Figuras*

FIGURA 1: DIAGRAMA DE CONEXIÓN A UN SISTEMA EXTERNO MEDIANTE LA CONEXIÓN SERIE .....	7
FIGURA 2:: ESQUEMA DE PINS DEL MÓDULO MSP430 RF2500 .....	9



## 1 INTRODUCCIÓN

El sistema utilizado para realizar el prototipo consta de diversas partes con capacidad para realizar más funciones de las requeridas por nuestro sistema. Esas partes pueden ser utilizadas para añadir funciones nuevas tanto a nuestro sistema como para ser utilizadas por otros sistemas.

Este documento detalla la capacidad de ampliación del sistema desarrollado, detallando todo aquello que puede ser utilizado para la conexión o el añadido de nuevos sistemas que trabajen conjuntamente con nuestro sistema.



## 2 OBJETIVO

Detallar la capacidad de ampliación de nuestro sistema.

### 3 CAPACIDAD DE AMPLIACIÓN

#### 3.1 Sistema de Alimentación

El sistema de alimentación da una tensión de salida de 3.3 V gracias a un regulador LDO. Se puede conectar en paralelo cualquier sistema que requiera una tensión de 3.3 V para funcionar, siempre controlando el consumo de corriente. El regulador LDO permite el paso de un máximo de 250 mA [1]. En la tabla 1 se ve la capacidad utilizada y disponible del sistema.

	Consumo máximo (A)
WIZ810MJ [2]	185,702 mA
MSP430 RF2500 [3]	0,390 mA
Total Utilizado	185,462 mA
Total Disponible	64,538 mA

Tabla 1: Consumo de corriente utilizado y disponible

#### 3.2 Conexión Serie

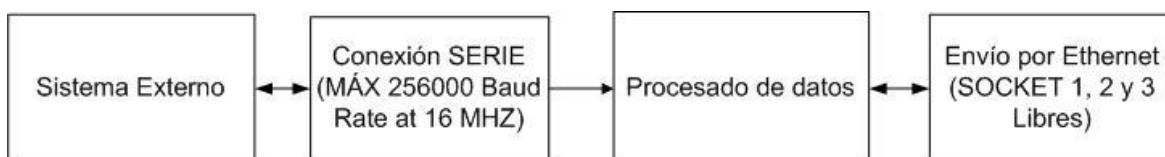


Figura 1: Diagrama de conexión a un sistema externo mediante la conexión SERIE



El sistema está configurado para emitir por serie la información de la posición de la superficie. Sin embargo, la conexión serie puede reprogramarse para poder ser utilizada como sistema de conexión a otro sistema, y enviar y recibir, conjuntamente con el módulo de ETHERNET WIZ810MJ, la información deseada. Las características del puerto serie del módulo MSP430 RF2500 pueden verse en la tabla 2:

Baudios por segundo (Reloj a 32 kHz)	{1200,9600}
Baudios por segundo (Reloj 1-16 MHz)	{9600,256000}
Tamaño de la información	7-8 Bits
Bits de paridad (opcional)	0,1
Selección de bits de parada	1 (por defecto) o 2

Tabla 2: Características de la conexión serie del módulo MSP430 RF2500

### 3.3 Conexión ETHERNET

El modulo de ETHERNET WIZ810MJ que funciona con el chip W5100 es un sistema que funciona mediante la programación de SOCKETS. Los sockets permiten la transmisión de datos de forma controlada y ordenada entre dos sistemas. Los sockets se pueden programar en función de diferentes protocolos. Los protocolos que pueden ser utilizados por el módulo son:

- TCP
- UDP
- ICMP
- IPv4 ARP
- IGMP
- PPPoE
- Ethernet

El módulo WIZ810MJ tiene capacidad de funcionar con 4 SOCKETS de manera simultánea. Uno está utilizado para el propio sistema. Por tanto, quedan libres para su utilización 3 sockets. El socket utilizado por el sistema propio es el 0, por tanto, quedan libres los sockets 1, 2 y 3. La programación de estos Sockets deberá llevarse a cabo a través del software del microcontrolador del sistema MSP430.

### 3.4 Procesado de datos (MSP430 RF2500):

El módulo MSP430 RF2500 que utiliza el microcontrolador MSP430 2274 tiene muchas características sin utilizar. Se pueden diferenciar las conexiones externas (Pins) y piezas de hardware que complementan la placa (leds, sensor de temperatura, etc.) de las funciones extra de software que se pueden utilizar.

En la **¡Error! No se encuentra el origen de la referencia.** se puede ver la distribución de pins en el módulo MSP430 RF2500.

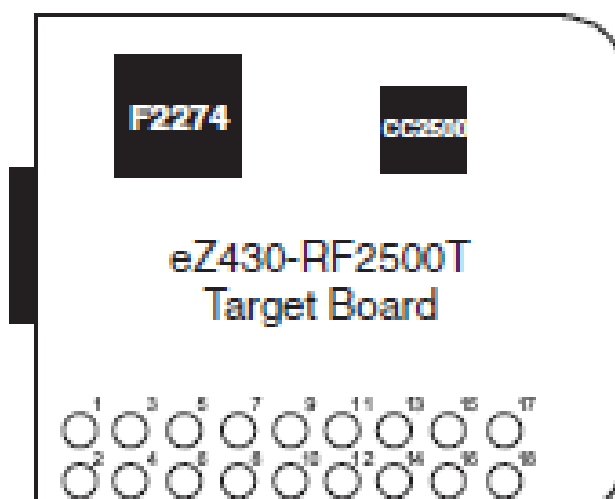


Figura 2: : Esquema de Pins del módulo MSP430 RF2500

#### 3.4.1 Funcionalidades de los pins libres

##### Pin 2

Salida de tensión de 3.3 V

##### Pin 5

Entrada y salida digital para uso general

Entrada analógica A2 para el conversor analógico digital ADC10

Entrada del Timer\_A CCI0B para el modo Capture. Este modo permite almacenar eventos temporales

Entrada del Timer\_A *Bootstrap Loader* (BSL). El *Bootstrap loader* es un sistema que permite al usuario programar tanto la memoria flash como la RAM usando una interfaz en serie UART

Salida de comparación del Timer\_A OUT0

Entrada y salida de reloj del Timer\_A TAO

#### Pin 6

Entrada y salida digital para uso general

Entrada del Timer\_A CCI1B para el modo Capture. Este modo permite almacenar eventos temporales

Salida de comparación del Timer\_A OUT1

Entrada analógica A3 para el amplificador operacional ADC10

Entrada y salida de tensión de referencia negativa

Entrada y salida de reloj del Timer\_A TA1

#### Pin 7

Entrada y salida digital para uso general

Salida de comparación del Timer\_A OUT2

Entrada analógica A4 para el amplificador operacional ADC10

Entrada y salida de tensión de referencia positiva

Entrada y salida de reloj del Timer\_A TA2

#### Pin 8

Entrada y salida digital para uso general

Entrada analógica A12 para el amplificador operacional ADC10

Entrada del Timer\_B CCI0B para el modo Capture. Este modo permite almacenar eventos temporales

Salida de comparación del Timer\_B OUT0

Entrada y salida de reloj del Timer\_B TB0

#### Pin 9

Entrada y salida digital para uso general

Entrada analógica A13 para el amplificador operacional ADC10

Entrada del Timer\_B CCI1B para el modo Capture. Este modo permite almacenar eventos temporales

Salida de comparación del Timer\_B OUT1

Entrada y salida de reloj del Timer\_B TB1

#### Pin 12

Toma de tierra

#### Pin 13

Entrada y salida digital para uso general

Terminal de entrada de un cristal oscilador

#### Pin 14

Entrada y salida digital para uso general  
Terminal de salida de un cristal oscilador

### 3.4.2 Hardware adicional

#### Conexión inalámbrica

Transceptor de Radio-Frecuencia a 2.4 GHz  
*Data rate* programable de hasta 500 kbps

#### Botones:

1 Pushbutton

### 3.4.3 Características accesibles por SOFTWARE

Memoria Flash de 32 kb + 256 b

Memoria RAM de 1 kb

Sensor de temperatura interno

1 conversor analógico digital extra ADC12

Frecuencias de operación de hasta 16 MHz

Cuatro modos de reloj calibrados con un error de error  $\pm 1\%$  :

- 2 MHz
- 4 MHz
- 8 MHz
- 16 MHz

Detector de cortes de corriente (*Brownout Detector*)

## 4 EJEMPLOS DE AMPLIACIÓN

El sistema se puede ampliar de diversas formas. En este apartado se exponen algunos ejemplos posibles que podrían utilizarse o aprovechar al sistema.

### 4.1 Monitorización de otras magnitudes

Mediante el uso de las diversas entradas analógicas y los conversores ADC, y de la misma forma que funciona el sistema desarrollado, sería posible conectar otros sensores que dieran una respuesta en tensión y que nos diera la información del sistema que se está midiendo. Algunos sensores posibles serían:

- Sensores de infrarrojos
- Sensores de ultrasonidos
- Sensores térmicos
- Sensores de presión
- Sensores de posición lineal o angular
- Etc.

### 4.2 Monitorización de la señal recibida por el servo

Se podría añadir la funcionalidad de no solo medir la posición en la que se encuentra el servo, sino también monitorizar la señal que ha recibido el servo. Sería una forma de controlar que la señal que se ha enviado para realizar y la posición del servo coinciden. Para realizar esto, habría que configurar una entrada en modo CAPTURE que es el modo que permite controlar variaciones de la señal en el tiempo y lo almacena.

### 4.3 Conexión a un POD subalar

La conexión de un POD subalar (sistema que se está desarrollando en otro grupo de trabajo) sería posible mediante la utilización de la conexión SERIE de la que dispone el sistema. Para la transmisión de datos podría utilizar fácilmente un SOCKET para realizar la transmisión de los datos a través de la red Ethernet del sistema.



#### 4.4 Monitorización del voltaje

Una función posible es monitorizar la tensión que llega al servo (o a otro sistema). De esta forma se puede controlar que esté funcionando correctamente y en caso de una caída de tensión importante, el sistema puede advertir de la situación que está sucediendo.



## 5 BIBLIOGRAFÍA

- [1] *A Low Dropout Positive Voltage Regulator Product Brief*, Ref. MCP1702, MICROCHIP Technology Inc., 2005
- [2] *W5100 Datasheet v. 1.1.8*, W5100\_Datasheet\_v1\_1\_8.pdf, WIZNET Inc., 4/03/2009
- [3] *MSP430x22x2, MSP430x22x4 MIXED SIGNAL MICROCONTROLLER*, msp430f2274.pdf, Texas Instruments Inc. Mayo de 2009



## 6 GLOSARIO

**LDO:** Low Drop Out

**TCP:** Transmission Control Protocol

**UDP:** User Datagram Protocol

**ICMP:** Internet Control Message Protocol

**IPv4:** Internet protocol version 4

**ARP:** Address Resolution Protocol

**IGMP:** Internet Group Management Protocol

**PPPoE:** Point-to-Point Protocol over Ethernet

**ADC:** Analog to Digital Converter





# Diseño de Sistema Sensor Para superficies de UAV

---

Iss./Rev.: D01

Date: 07/07/2009

[END OF DOCUMENT]

## ANEXO 6: Presupuesto

	Name	Signature	Date
Proyectista 1	Eloy Herrera		07/07/09
Proyectista 2	Víctor de los Santos		07/07/09
Director	Jorge Ramírez Alcantara		07/07/09

### *Registro de cambios*

Fecha	Cambio
08/07/09	-Se añade el presupuesto final y se hacen los cambios pertinentes

## CONTENIDO

<b>1</b>	<b>INTRODUCCIÓN .....</b>	<b>5</b>
<b>2</b>	<b>OBJETIVO .....</b>	<b>6</b>
<b>3</b>	<b>PRESUPUESTO INICIAL .....</b>	<b>7</b>
<b>4</b>	<b>PRESUPUESTO FINAL.....</b>	<b>8</b>
<b>5</b>	<b>BIBLIOGRAFIA .....</b>	<b>9</b>

## *Índice de Tablas*

TABLA 1: PRESUPUESTO INICIAL .....	7
TABLA 2: PRESUPUESTO FINAL .....	8



## 1 INTRODUCCIÓN

La realización de un proyecto de ingeniería conlleva unos gastos que deben ser totalmente reportados y aceptados por el director de proyecto. Este documento contiene el presupuesto propuesto y definitivo del proyecto



## 2 OBJETIVO

Realizar el presupuesto del proyecto

### 3 PRESUPUESTO INICIAL

El presupuesto inicial fue el presupuesto propuesto para la adquisición de los productos. Este presupuesto se realizó utilizando la tienda online RS-ONLINE (AMIDATA), Tienda 2000 y con un diseño preliminar. La Tabla 1 muestra el presupuesto inicial.

Tabla 1: Presupuesto Inicial

Producto	Referencia	Precio/Unidad (€)	Nº Unidades	Precio Total (€)
Regulador	LM2397 ET [1]	1,08	5**	5,40
Módulo Ethernet	WIZ810MJ [2]	26,33	1	26,33
Módulo Micropro- cesador	MSP430 RF2500 [3]	43,89	1	43,89
Condensador	0,1 uF [4]	0,17	5**	0,85
Condensador	10 uF [5]	0,47	5**	2,35
Servo	Futaba S3003* [6]	11,00	1	11,00
	<b>Precio (Sin Ser- vo)</b>			<b>78.85 €</b>
	<b>Precio (Con Ser- vo)</b>			<b>89.85 €</b>
	<b>PRECIO DE UN SISTEMA POR SUPERFICIE (no lleva servo)</b>	78,85	4	<b>TOTAL = 315,4€</b>

\*Hay que añadir un SERVO de PRUEBAS para poder realizar el prototipo antes de trabajar con los servos del UAV. El servo elegido es el Futaba S3003 debido a que la plataforma de pruebas es de la marca FUTABA y de esta forma se asegura la compatibilidad con el servo.

\*\* Compra mínima



## 4 PRESUPUESTO FINAL

Tras una revisión del diseño detallado, y además debido a falta de stock en la tienda inicial, el presupuesto final es el que aparece en la Tabla 2. Además, se cambia el regulador y sus condensadores de acondicionamiento debido a que el anterior no cumplía todos los requisitos necesarios. Las tiendas donde se han comprado los componentes son: RS-ONLINE, Farnell, y Planet Hobby S.L. El resultado es un presupuesto ligeramente superior que el calculado inicialmente.

Tabla 2: Presupuesto final

Producto	Referencia	PVP/Unidad (€)	Nº Unidades	PVP Total (€)
Regulador	MCP1702 ET [7]	0,78	1	0,78
Módulo Ethernet	WIZ810MJ [2]	26,33	1	26,33
Módulo Micropro- cesador	MSP430 RF2500 [8]	50,51	1	50,51
Condensador 1 µF cerámico	CK06BX105K [9]	2,48	2	4,96
Servo	Futaba S3003 [10]	8,61	1	8,61
	<b>Precio (Sin Servo)</b>			<b>82,58 €</b>
	<b>Precio (Con Servo)</b>			<b>91,19 €</b>
	<b>PRECIO DE UN SISTEMA POR SUPERFICIE (no lleva servo)</b>	78,85	4	<b>TOTAL = 330,32 €</b>

Los precios son sin IVA y sin gastos de envío. Los gastos de envío ascienden a 10 euros (sin IVA).

## 5 BIBLIOGRAFIA

- [1] RS-ONLINE, 20/06/09, <http://es.rs-online.com/web/search/searchBrowseAction.html?method=getProduct&R=5358950>
- [2] RS-ONLINE, 20/06/09, <http://es.rs-online.com/web/search/searchBrowseAction.html?method=getProduct&R=0419895>
- [3] RS-ONLINE, 20/06/09, <http://es.rs-online.com/web/search/searchBrowseAction.html?method=searchProducts&searchTerm=MSP430+RF2500&x=0&y=0>
- [4] RS-ONLINE, 20/06/09, <http://es.rs-online.com/web/search/searchBrowseAction.html?method=getProduct&R=2218584>
- [5] RS-ONLINE, 20/06/09, <http://es.rs-online.com/web/search/searchBrowseAction.html?method=getProduct&R=2218685>
- [6] Tienda 2000, 20/06/09, Tienda 2000\*, <http://www.tienda2000.com/servos.htm>
- [7] FARNELL, 30/06/09, <http://es.farnell.com/jsp/search/productdetail.jsp?SKU=1331485>
- [8] FARNELL, 30/06/09, <http://es.farnell.com/jsp/search/productdetail.jsp?SKU=1382267>
- [9] FARNELL, 30/06/09, <http://es.farnell.com/jsp/search/productdetail.jsp?SKU=1650918>
- [10] Tienda Planet Hobby S.L., 08/07/09, Barcelona



Diseño de un  
Sistema de  
Monitorización de  
configuración de  
UAV

Iss./Rev.: D01

Date: 07/07/2009

---

[END OF DOCUMENT]

# ANEXO 7: Functional Hazard Assessment (FHA)

	Name	Signature	Date
Proyectista 1	Eloy Herrera		12/07/09
Proyectista 2	Víctor de los Santos		12/07/09
Director	Jorge Ramírez Alcantara		12/07/09

## Registro de cambios

Fecha	Cambio
10/07/09	-Se cambia la introducción por otra que ofrece una guía más detallada de cómo elaborar un PSSA
13/07/09	- Se cambia el formato.

<b>1</b>	<b>INTRODUCCIÓN .....</b>	<b>6</b>
1.1	FHA Nivel Avión: .....	6
1.2	FHA Nivel sistema .....	7
<b>2</b>	<b>IDENTIFICACIÓN DE LAS FUNCIONES .....</b>	<b>10</b>
2.1	Introducción .....	10
2.2	Desarrollo: .....	10
2.3	Lista de funciones .....	11
<b>3</b>	<b>IDENTIFICACIÓN Y DESCRIPCIÓN DE LAS CONDICIONES DE FALLO.....</b>	<b>12</b>
3.1	Introducción .....	12
3.2	Desarrollo .....	12
<b>4</b>	<b>EFFECTOS DE LAS CONDICIONES DE FALLO .....</b>	<b>13</b>
4.1	Introducción .....	13
4.2	Desarrollo: .....	13
<b>5</b>	<b>CLASIFICACIÓN DE LOS EFECTOS DE LAS CONDICIONES DE FALLO.....</b>	<b>14</b>
5.1	Introducción: .....	14
5.2	Desarrollo: .....	14
<b>6</b>	<b>ASIGNACIÓN DE LOS REQUISITOS DE PROBABILIDAD A LAS CONDICIONES DE FALLO .....</b>	<b>15</b>
<b>7</b>	<b>BIBLIOGRAFÍA .....</b>	<b>16</b>
<b>8</b>	<b>GLOSARIO .....</b>	<b>17</b>

TABLA 1: OBJETIVOS DEL SISTEMA Y SUS CONDICIONES DE FALLO .....	11
TABLA 2: ASIGNACIÓN DE PROBABILIDADES DE FALLO A LAS DIFERENTES CONDICIONES DE FALLO.....	14
TABLA 3: REQUISITOS DE PROBABILIDAD DE NUESTRO SISTEMA.....	15

## *Índice de Figuras*

FIGURA 1: EJEMPLO DE CONDICIONES DE FALLO PARA UNA FUNCIÓN.....	6
FIGURA 2: FUNCIONES DE ALTO NIVEL DE LA AERONAVE.....	7
FIGURA 3: ESQUEMA DEL PROCESO DE UN FHA A NIVEL DE SISTEMA.....	8
FIGURA 4: INTEGRACIÓN ENTRE LAS CONDICIONES DE FALLO DE NUESTRO SISTEMA .....	13



## 1 INTRODUCCIÓN

Los objetivos del “Functional Hazard Assessment” (FHA) es coger las distintas funciones y identificar sus condiciones de fallo y su clasificación según impliquen una pérdida de la función o simplemente un malfuncionamiento de ésta. En el FHA también fijaremos los requisitos de safety necesarios para limitar los efectos del fallo de una función que puedan afectar a la clasificación de la condición de fallo como podemos observar en el ejemplo de la **Figura 3.1**.

Existen dos niveles dentro de un FHA, el nivel avión y el nivel sistema.

Function	Failure Condition
Control Flight Path	Inability to control Flight Path
Control Touchdown and Roll Out	Inability to control Touchdown and Roll Out
Control Thrust	Inability to control Thrust
Control Cabin Environment	Inability to control Cabin Environment
Provide Spatial Orientation	Inability to provide Spatial Orientation
Fire Protection	Loss of Fire Protection

Figura 1: Ejemplo de condiciones de fallo para una función

### 1.1 FHA Nivel Avión:

Para el nivel avión usaremos un estudio de safety desarrollado por la agencia espacial americana NASA. Como conclusión más importante de este informe de seguridad podemos tomar que la función que implemente nuestro sistema no es crítica como podemos apreciar en la **¡Error! No se encuentra el origen de la referencia..**

Esto tiene una implicación directa en el DAL a asumir, ya que ni la condición de fallo más grave que contemplamos (la pérdida del sistema) tiene efectos en el safety.

En principio, y hasta que estudios posteriores puedan decir lo contrario, al no tocar ninguna de las funciones de más alto nivel del avión podemos considerar que nuestro sistema no tiene efecto alguno en el safety del avión.

Para información más detallada acerca de las funciones y su criticidad dentro del sistema aeronave consultar el documento “NASA-2007-214539”

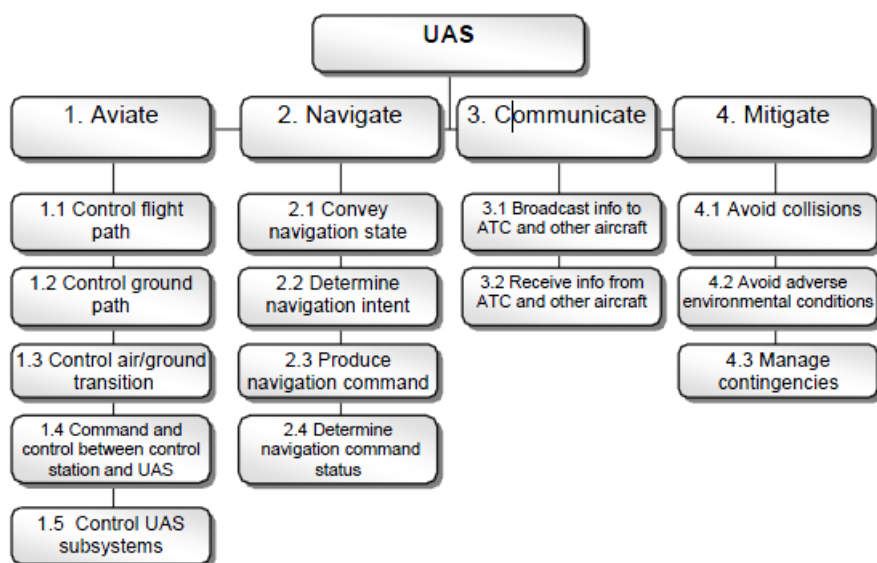


Figura 2: Funciones de alto nivel de la aeronave

## 1.2 FHA Nivel sistema

El análisis a nivel avión sobretodo se utiliza para identificar posibles combinaciones de fallos que acaben superando la clasificación de las condiciones de fallo establecidas anteriormente.

Para la realización de un FHA no deberemos utilizar documentos adicionales ya que éste constituye el primer paso en un proceso de safety.

En el proceso para identificar las condiciones de fallo y sus efectos deberemos seguir todo un conjunto de subprocesos que podemos ver de forma esquemática en la **Figura 3**:

A) Identificar todas las funciones del sistema a estudiar.

Para identificar una función deberemos obtener primero toda la información necesaria, esto es, en un FHA a nivel de sistema:

- La lista de las funciones principales a considerar
- Un diagrama con las interfaces exteriores
- Una lista de las funciones de alto nivel descritas en el FHA
- Una lista de las condiciones de fallo de alto nivel descritas en el FHA
- Los requisitos definidos por el departamento de diseño y el resto de la documentación.
- La opción de diseño escogida y su justificación.

B) Identificar y describir las condiciones de fallo

Este proceso empieza con la elaboración de una lista de configuración de emergencia que contemple inclemencias meteorológicas, presencia de volcanes, etc. Después utilizaremos esta lista junto con la elaborada en el proceso anterior para elaborar otra lista que incluya todas las condiciones de fallo posibles tales como una pérdida del sistema hidráulico o eléctrico, la congelación del sistema, etc. Para la elaboración de esta lista deberemos antes entender los modos de

fallo.

C) Determinar los efectos de las condiciones de fallo

Los analistas deben determinar los efectos de las condiciones de fallo en el avión. Para esto se basaran en el criterio de personal con experiencia en el sector.

D) Determinar la clasificación de los efectos de las condiciones de fallo.

Se refiere a la clasificación del sistema dentro de cualquiera de estos niveles:

- Catastrophic
- Severe-major
- Major
- Minor
- No safety effect

E) Asignar requisitos posibles derivados de estas condiciones de fallo

Para cada condición de fallo el analista deberá asignar posibles requisitos del sistema y requisitos de diseño.

F) Identificar el método usado para verificar el cumplimiento de los requisitos derivados de las condiciones de fallo. Para cada condición de fallo el analista deberá explicar como la aeronave satisfará este requisito.

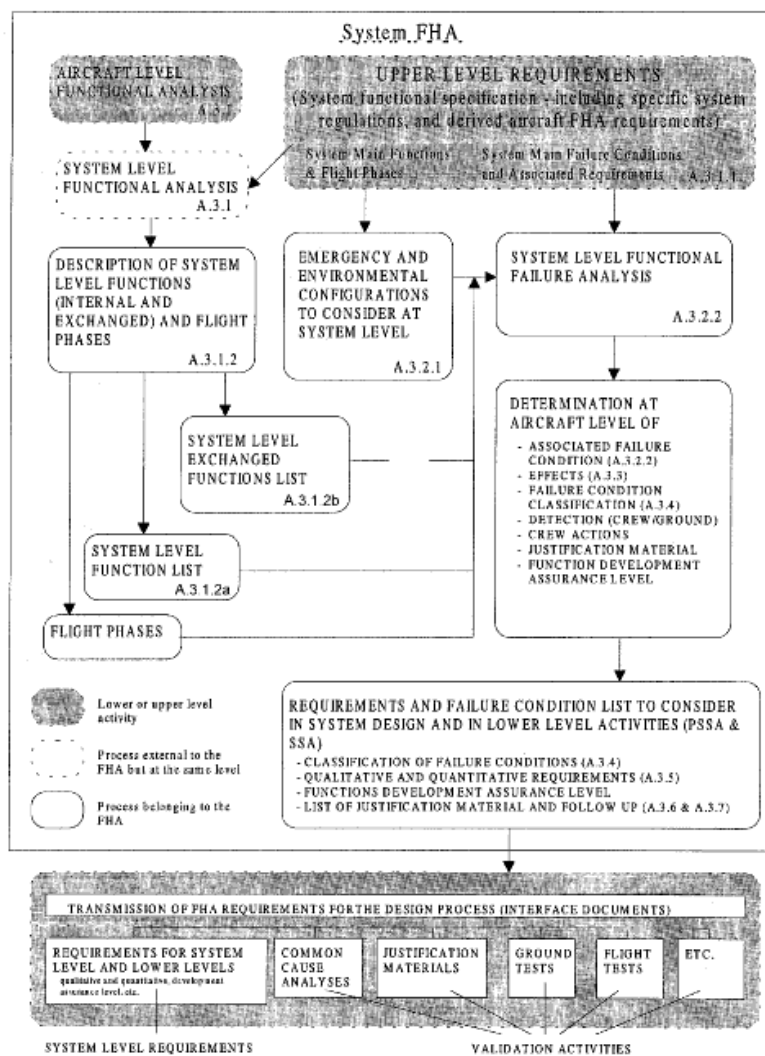


Figura 3: Esquema del proceso de un FHA a nivel de sistema



## 2 OBJETIVO

Realizar el Functional Hazard Assessment (FHA) del sistema sensor que estamos diseñando.

## 3 IDENTIFICACIÓN DE LAS FUNCIONES

### 3.1 Introducción

Según esta recogido en el documento ARP4761, todas las funciones asociadas con el sistema deben ser identificadas en este punto. Identificar una función se trata de obtener toda la información relativa a ella y con ésta crear una lista de funciones.

Como información relativa a la función debemos tener las siguientes entradas para realizar el FHA de nuestro sistema:

- a) Una lista de las funciones de alto nivel de nuestra aeronave.
- b) Los objetivos de la aeronave y los requisitos del cliente
- c) Las decisiones iniciales de diseño

Una vez reunida toda la información necesaria pasaremos a crear la lista de funciones. Esta lista se hace tomando como referencia la lista de funciones esperada y teniendo en cuenta la información relativa a la función recopilada en el apartado anterior.

Durante este proceso encontraremos dos tipos de funciones:

- a) **Funciones internas:** Son aquellas funciones principales del avión proporcionadas o llevadas a cabo por los sistemas internos de la aeronave. Cuando el safety es de un sistema, como es nuestro caso, las funciones internas serán aquellas funciones propias del sistema y llevadas a cabo por el equipo interno de éste.
- b) **Funciones externas:** En un avión estas funciones son las relativas a la interoperación con otras aeronaves o con tierra. A nivel sistema, estas funciones están dadas por la interconexión de nuestro sistema con otros sistemas embarcados en el avión.

### 3.2 Desarrollo:

Información relativa a la función:

a.) Lista de funciones

- Función 1: El sistema mostrará el desplazamiento angular del alerón izquierdo
- Función 2: El sistema mostrará el desplazamiento angular del alerón derecho
- Función 3: El sistema mostrará el desplazamiento angular del timón de profundidad
- Función 4: El sistema mostrará el desplazamiento angular del timón de dirección
- Función 5: El sistema mostrará las revoluciones del motor.

b.)

Muestreo del desplazamiento angular del alerón izquierdo	Fallo a la hora de mostrar el desplazamiento angular del alerón izquierdo
Muestreo del desplazamiento angular del alerón derecho	Fallo a la hora de mostrar el desplazamiento angular del alerón derecho
Muestreo del desplazamiento angular del timón de profundidad	Fallo a la hora de mostrar el desplazamiento angular del timón de profundidad
Muestreo del desplazamiento angular del timón de dirección	Fallo a la hora de mostrar el desplazamiento angular del timón de dirección
Muestreo de las revoluciones del motor	Fallo a la hora de mostrar las revoluciones del motor

Tabla 1: Objetivos del sistema y sus condiciones de fallo

c.) Requisitos **R001, R002, R003 y R004**

### 3.3 Lista de funciones

a.) Funciones internas:

- 1- El sistema recogerá datos de los sensores del alerón izquierdo
- 2- El sistema recogerá datos de los sensores del alerón derecho
- 3- El sistema recogerá datos de los sensores del timón de profundidad
- 4- El sistema recogerá datos de los sensores del timón de dirección

b.) Funciones externas:

- 1- El sistema enviará los datos procesados por el sistema de comunicaciones del UAV.

## 4 IDENTIFICACIÓN Y DESCRIPCIÓN DE LAS CONDICIONES DE FALLO

### 4.1 Introducción

Según esta recogido en el documento ARP4761, el proceso de identificación de las condiciones de fallo empieza creando una descripción del entorno en el que trabajará la aeronave y una lista de configuraciones de emergencia. Después el analista deberá considerar todos los sistemas que intervienen en la lista de funciones internas, la lista de funciones externas, el entorno en el que trabajará nuestro sistema y la lista de configuraciones de emergencia. Con esto el analista creará una lista de las condiciones de fallo del sistema considerando fallos tanto puntuales como múltiples en un entorno normal y en un entorno alterado. Esta lista se deberá dividir en fases de vuelo cuando la criticidad del sistema así lo aconseje, no es nuestro caso ya que nuestro sistema es de DAL C

En la descripción del entorno en el que trabajará la aeronave y la lista de configuraciones de emergencia deberemos:

- a) Definir la climatología en la que nuestro sistema trabajará .
- b) Crear una lista de todas aquellas condiciones en que nuestro sistema vuele pero no lo haga en las condiciones nominales.

Cuando consideremos los efectos de un fallo puntual o un fallo múltiple deberemos crear dos listas:

- a) Fallo puntual del sistema: Deberemos considerar la lista de condiciones de fallo creada en el apartado anterior y el documento de diseño arquitectural (ANEXO 2) creado por el departamento de diseño.
- b) Fallo múltiple: Este apartado no se contempla en nuestro sistema al entender que cualquier fallo individual del sistema implicará una pérdida de éste. Si lo hubiéramos hecho deberíamos haber tenido en consideración la integración de los diferentes subsistemas entre ellos.

### 4.2 Desarrollo

#### **Descripción del entorno en el que trabajará la aeronave y la lista de configuraciones de emergencia:**

##### **a.) Climatología**

Despejado: El UAV solo se contempla para volar en condiciones de clima despejado y con poco viento. No volará en caso de que la climatología sea diferente.

##### **b.) Condiciones anómalas:**

- Pérdida del sistema eléctrico
- Pérdida del sistema de transmisión
- Pérdida de la alimentación del sistema
- Pérdida total del sistema eléctrico del UAV.

#### **Condiciones de fallo considerando fallos puntuales o múltiples:**

Fallo 1- Fallo a la hora de mostrar el desplazamiento angular del alerón izquierdo

Fallo 2- Fallo a la hora de mostrar el desplazamiento angular del alerón derecho

Fallo 3- Fallo a la hora de mostrar el desplazamiento angular del timón de profundidad

Fallo 4- Fallo a la hora de mostrar el desplazamiento angular del timón de dirección

## 5 EFECTOS DE LAS CONDICIONES DE FALLO

### 5.1 Introducción

Según esta recogido en el documento ARP4761, el analista debe determinar los efectos de las condiciones de fallo en la operación normal de la aeronave o la carga. Para esto podremos consultar la opinión de expertos y gente con experiencia en la materia para ayudarnos a clasificar los efectos de las condiciones de fallo. En el caso de estar desarrollando un análisis de safety para un sistema, como es nuestro caso, el efecto a nivel de avión puede ser el mismo que a nivel de sistema o podemos considerar la combinación de efectos de otros sistemas.

### 5.2 Desarrollo:

Fallo 1: Incapacidad de determinar la posición del alerón izquierdo, obliga a determinar su posición en función de la actitud del avión.

Fallo 2: Incapacidad de determinar la posición del alerón derecho, obliga a determinar su posición en función de la actitud del avión.

Fallo 3: Incapacidad de determinar la posición del timón de profundidad, obliga a determinar su posición en función de la actitud del avión.

Fallo 4: Incapacidad de determinar la posición del timón de dirección, obliga a determinar su posición en función de la actitud del avión.

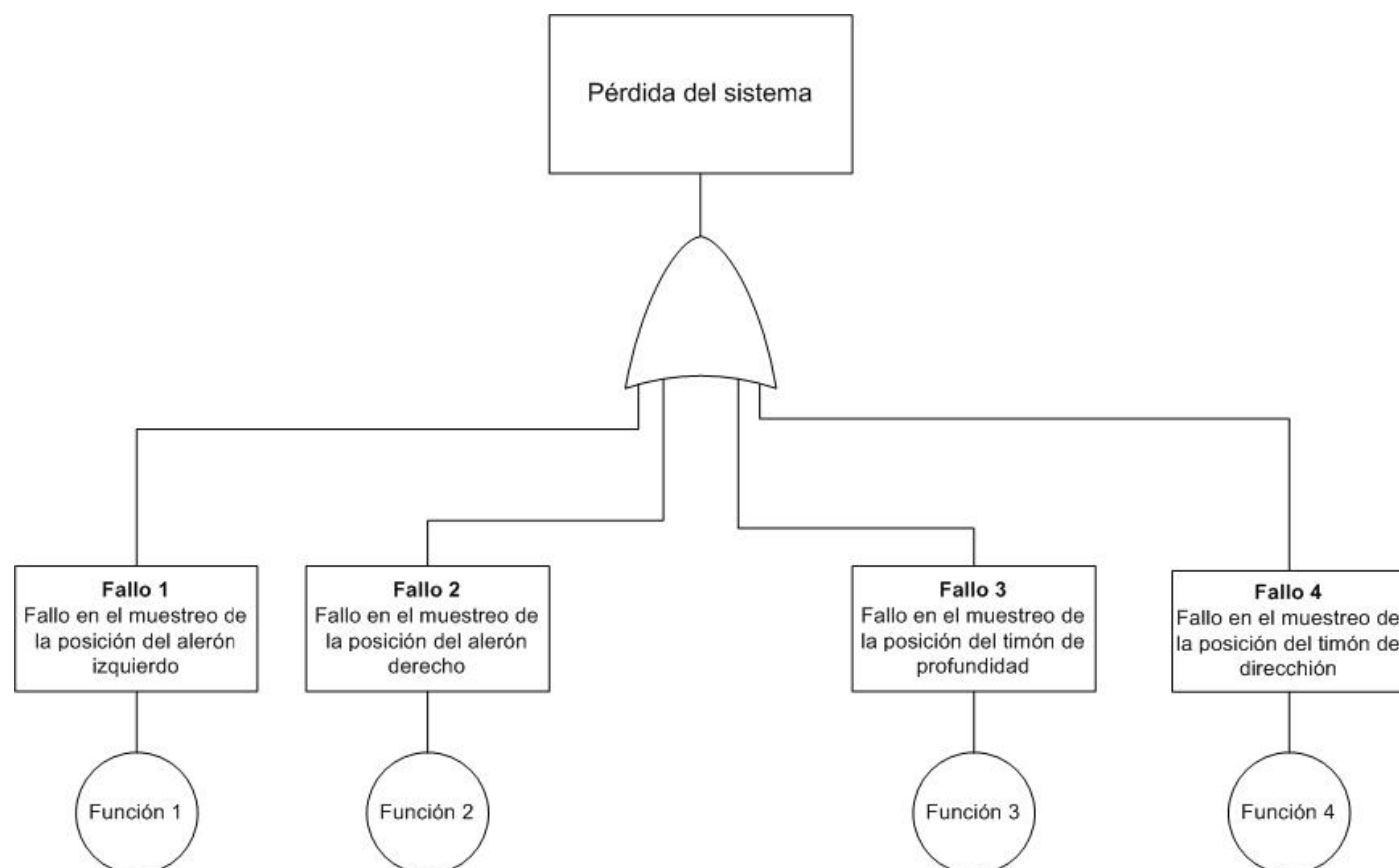


Figura 4: Integración entre las condiciones de fallo de nuestro sistema



## 6 CLASIFICACIÓN DE LOS EFECTOS DE LAS CONDICIONES DE FALLO

### 6.1 Introducción:

La determinación de esta clasificación viene dada por el análisis de los documentos o reportes de accidentes o incidente, la revisión del documento de quía regulado, usando experiencias de diseño anteriores y consultando con expertos y gente con experiencia en el campo en el que se mueve el sistema. Las clasificaciones según el anteriormente citado SAE ARP4761 son:

- Catastrophic
- Severe-Major
- Hazardous
- Major
- Minor
- No safety effect

### 6.2 Desarrollo:

El sistema tratado en este FHA es en general un sistema simple, que en principio no se iba a incluir en el UAV. Al añadirse finalmente, se ha realizado un estudio del sistema para decidir que clasificación debían tener los Failure Condition del FHA. En principio, al ser un sistema añadido que no estaba dentro de las previsiones del UAV, se consideró que los efectos de un fallo no provocaba ningún efecto de Safety en el avión, o que era un efecto menor. Sin embargo, al ser un sistema que todos los aviones reales están obligados a llevar, consideramos que el UAV necesitaría tarde o temprano de un sistema como el que se está desarrollando. A partir de ahí, se decidió que el sistema adquiriera una clasificación del Failure Condition Effect de nivel Mayor (System Development Assurance Level - C). Debido a la arquitectura del sistema, el sistema se considera que falla cuando cualquiera de los 5 posibles fallos de sistema aparece. Eso lleva a que cada fallo debe tener la misma clasificación del Failure Condition Effect. El resultado lo podemos ver en la **Tabla 2**.

Failure Condition	Failure Condition Classification	System Assurance Level	Development	Probability
Pérdida del sistema	Major	C		$P < 10\text{exp-}5$
Fallo 1	Major	C		$P < 10\text{exp-}5$
Fallo 2	Major	C		$P < 10\text{exp-}5$
Fallo 3	Major	C		$P < 10\text{exp-}5$
Fallo 4	Major	C		$P < 10\text{exp-}5$

Tabla 2: Asignación de probabilidades de fallo a las diferentes condiciones de fallo.

## 7 ASIGNACIÓN DE LOS REQUISITOS DE PROBABILIDAD A LAS CONDICIONES DE FALLO

Según esta recogido en el documento ARP4761, Para cada condición de fallo, el analista debe asignar un requisito de probabilidad que se reportarán a los departamentos encargados del proceso de diseño.

Los requisitos que nosotros hemos considerado están en la **Tabla 3**.

<p><b>[RS000] La probabilidad de pérdida del sistema debe estar clasificada como "Major Failure Condition"</b></p> <p>SW02-01</p> <p>FUENTE: Realización del FHA</p> <p>[Diseño arquitectural] [NO IMPLEMENTADO]</p>
<p><b>[RS0001] La probabilidad del fallo 1 debe estar clasificada como "Major Failure Condition"</b></p> <p>SW02-01</p> <p>FUENTE: Realización del FHA</p> <p>[Diseño arquitectural] [NO IMPLEMENTADO]</p>
<p><b>[RS002] La probabilidad del fallo 2 debe estar clasificada como "Major Failure Condition"</b></p> <p>SW02-01</p> <p>FUENTE: Realización del FHA</p> <p>[Diseño arquitectural] [NO IMPLEMENTADO]</p>
<p><b>[RS003] La probabilidad del fallo 3 debe estar clasificada como "Major Failure Condition"</b></p> <p>SW02-01</p> <p>FUENTE: Realización del FHA</p> <p>[Diseño arquitectural] [NO IMPLEMENTADO]</p>
<p><b>[RS004] La probabilidad del fallo 4 debe estar clasificada como "Major Failure Condition"</b></p> <p>SW02-01</p> <p>FUENTE: Realización del FHA</p> <p>[Diseño arquitectural] [NO IMPLEMENTADO]</p>

**Tabla 3: Requisitos de probabilidad de nuestro sistema.**

Según el documento CS-23, la probabilidad para un "Major Failure Condition" será de 1 fallo cada 100000 horas de vuelo.

La atribución de esta clasificación se ha llevado a cabo a través de un estudio de los tipos de clasificaciones de los errores según el documento CS-23. Según este documento, una "Major Failure Condition" debe ser asignada siempre que la condición de fallo pueda reducir la capacidad del avión o la habilidad de la tripulación para tratar con condiciones con condiciones de operación adversas, como por ejemplo una reducción de los márgenes de seguridad o de capacidades funcionales, un incremento considerable de trabajo para la tripulación o reduciendo la eficiencia de esta, etc. En este caso, usamos un documento pensado para aeronaves pequeñas y de aviación general, debido a que no existe todavía ningún documento que trate directamente con temas de UAV (aunque se está desarrollando). Sin embargo, a pesar de que el UAV es un sistema no tripulado, hay alguien que está pendiente y siguiendo los parámetros que va recibiendo constantemente del UAV. Eso nos lleva a asignarle una clasificación "Major" ya que la información transmitida es importante y la ausencia de esta puede llevar a un aumento considerable de la carga de trabajo del que esté controlando desde tierra el UAV.



## 8 BIBLIOGRAFÍA

- SAE ARP4761, "Guidelines and methods for conducting the safety assessment process on civil airborne systems and equipment". U.S.A., 1996



## 9 GLOSARIO

FHA: Functional Hazard Assessment

PSSA: Preliminary System Safety Assessment

ZSA: Zonal Safety Assessment

FTA: Fault Tree Analysis



# Diseño de Sistema Sensor Para superficies de UAV

---

Iss./Rev.: 12/07/2009

Date: 07/07/2009

[END OF DOCUMENT]

# ANEXO 8: Preliminary System Safety Assessment (PSSA)

	Name	Signature	Date
Proyectista 1	Eloy Herrera		12/07/09
Proyectista 2	Víctor de los Santos		12/07/09
Director	Jorge Ramírez Alcantara		12/07/09

## Registro de cambios

Fecha	Cambio
10/07/09	-Se cambia la introducción por otra que ofrece una guía más detallada de cómo elaborar un PSSA
12/07/09	- Se cambia el formato. -Se añaden requisitos -Se añaden nuevos gráficos

<b>1</b>	<b>INTRODUCCIÓN .....</b>	<b>6</b>
<b>2</b>	<b>LISTA DE REQUISITOS DE “SAFETY” DEL AVIÓN .....</b>	<b>8</b>
<b>3</b>	<b>EVALUAR LAS DECISIONES DE DISEÑO CON RESPECTO A LOS REQUISITOS Y OBJETIVOS DEL SISTEMA.....</b>	<b>17</b>
<b>4</b>	<b>DERIVAR LOS REQUISITOS DE “SAFETY” AL DISEÑO DE LOS SUBSISTEMAS.....</b>	<b>20</b>
4.1	LISTA DE CONDICIONES DE FALLO ACTUALIZADA QUE INCLUYA UNAS DIRECTRICES DE CÓMO LOS REQUISITOS PUEDEN SER SATISFECHOS.....	20
4.2	LISTA CON LOS REQUISITOS DE “SAFETY” INTERRELACIONADOS CON EL HARDWARE Y SOFTWARE. ....	21
4.3	REQUISITOS PARA LA INSTALACIÓN DEL SISTEMA .....	22
4.4	DAL DEL SOFTWARE Y EL HARDWARE .....	23
4.5	TAREAS DE MANTENIMIENTO .....	24
<b>5</b>	<b>BIBLIOGRAFÍA .....</b>	<b>25</b>
<b>6</b>	<b>GLOSARIO .....</b>	<b>26</b>



TABLA 1: REQUISITOS PARA LAS CONDICIONES DE FALLO DEL ALERÓN IZQUIERDO. ....	10
TABLA 2: REQUISITOS PARA LAS CONDICIONES DE FALLO DEL ALERÓN DERECHO .....	12
TABLA 3: REQUISITOS PARA LAS CONDICIONES DE FALLO DEL TIMÓN DE PROFUNDIDAD.....	14
TABLA 4: REQUISITOS PARA LAS CONDICIONES DE FALLO DEL TIMÓN DE DIRECCIÓN.....	16
TABLA 5: CLASIFICACIÓN DE LAS CONDICIONES DE FALLO SEGÚN SU DAL.....	19
TABLA 6: CONDICIONES DE FALLO ACTUALIZADAS .....	21
TABLA 7: REQUISITOS DE “SAFETY” INTERRELACIONADOS CON EL HARDWARE Y SOFTWARE .....	22
TABLA 8: REQUISITOS PARA LA INSTALACIÓN DEL SISTEMA. ....	23
TABLA 9: CLASIFICACIÓN DE LAS CONDICIONES DE FALLO SEGÚN SU DAL.....	23
TABLA 10: DAL DE SOFTWARE Y HARDWARE .....	24

## *Índice de Figuras*

FIGURA 1: CONDICIONES DE FALLO PARA EL ALERÓN IZQUIERDO .....	9
FIGURA 2: CONDICIONES DE FALLO PARA EL ALERÓN DERECHO.....	11
FIGURA 3: CONDICIONES DE FALLO PARA EL TIMÓN DE PROFUNDIDAD.....	13
FIGURA 4: CONDICIONES DE FALLO PARA EL TIMÓN DE DIRECCIÓN. ....	15
FIGURA 5: DIAGRAMA DE PÉRDIDA DEL SISTEMA .....	17
FIGURA 6: ARQUITECTURA DEL SISTEMA .....	18
FIGURA 7: DAL DE LOS DIFERENTES SUBSISTEMAS.....	19

## 1 INTRODUCCIÓN

Según esta recogido en el documento ARP4761, el “Preliminary System Safety Assessment” (PSSA) consiste en examinar la arquitectura del sistema propuesto determinando como los fallos en ésta nos pueden llevar a los peligros identificados en el “Functional Hazard Assessment” (FHA) y cómo los requisitos del FHA pueden ser identificados. El proceso del PSSA interactúa y está asociado con el de diseño, yendo los dos unidos a través del proceso iterativo que es una espiral de diseño.

Para cada sistema analizado el PSSA reportará un conjunto de condiciones de fallo que pueden llevarnos a los peligros descritos en el FHA.

Para la elaboración de un PSSA deberemos tener en cuenta el diseño de la arquitectura del sistema, su complejidad, la peligrosidad de sus condiciones de fallo y el tipo de función que nuestro sistema realiza dentro de la aeronave. Para esto, un requisito primordial será el FHA.

Para determinar cómo los fallos en el sistema pueden llevar a los peligros destacados en el FHA y como los requisitos del FHA pueden ser satisfechos, deberemos completar una lista con los requisitos de safety del sistema. Para la elaboración de ésta lista necesitaremos como documentos de entrada:

- FHA
- Un Common Cause Analysis (CCA) provisional.
- Una descripción de la arquitectura del sistema (y su justificación).
- Una lista del equipo a utilizar y sus respectivas funciones.
- Las relaciones entre los distintos sistemas.

Después deberemos determinar si nuestra arquitectura y/o diseño permite cumplir con estos requisitos. Para esto deberemos tener en cuenta:

- Un FTA dónde se muestre como los diferentes fallos por si mismos o mediante una combinación de ellos pueden llegar a cumplir con la condición de fallo.
- Una identificación de posibles independencias entre los fallos.
- Un FTA dónde se muestre que requisitos y objetivos relacionados con las condiciones de fallo se pueden verificar y cuáles no.
- La determinación de los procesos y tiempos de mantenimiento de los diferentes subsistemas.
- La determinación de un DAL para los diferentes subsistemas.

Y por último trazar estos requisitos al diseño de los pequeños subsistemas que forman nuestro sistema (software y hardware). Para esto deberemos realizar:

- Una lista de las condiciones de fallo actualizadas.
- Los requisitos de safety para cada subsistema.
- Los DALs del software y el hardware.
- Determinar las tareas de mantenimiento y sus tiempos.



## 2 OBJETIVO

Realizar el Preliminary System Safety Assessment (PSSA) del sistema sensor que estamos diseñando.



### 3 LISTA DE REQUISITOS DE “SAFETY” DEL AVIÓN

Según esta recogido en el documento ARP4761, en la redacción del FHA creemos una lista de funciones a las que atribuimos unas condiciones de fallo y unos requisitos derivados de safety para estas condiciones de fallo. Ahora en el PSSA deberemos retomar estas condiciones de fallo primarias del FHA y desgranarlas en condiciones de fallo de los diferentes subsistemas.

Adjuntaremos también un *Fault tree analysis* (Consultar ARP4761 apartado 4.1 para una explicación detallada de cómo llevarlo a cabo) donde mostraremos como las distintas condiciones de fallo desgranadas en el PSSA pueden llegar a producir la condición de fallo mayor proveniente del FHA.

Deberemos también completar unos requisitos de safety para estos subsistemas que nos aseguren el no incurrir en ninguna de las condiciones de fallo destapadas en el PSSA. Reportaremos al departamento de diseño estos requisitos para que los tengan en cuenta en el proceso de diseño del sistema.

#### Condición de fallo del FHA:

*Fallo a la hora de mostrar el desplazamiento angular del alerón izquierdo.*

Requisitos heredados:

[RS001]

Condiciones de fallo:

- **Fallo 1: Fallo en la recepción de datos del sensor.**  
El valor de voltaje que tomamos del potenciómetro es incorrecto.
- **Fallo 2: Fallo en el procesamiento de señal realizado por el microprocesador.**  
El microprocesador corrompe los datos tomados por el sensor.
- **Fallo 3: Fallo en la transmisión de la señal hasta la emisora del "Shadow". Cableado.**  
La señal transportada por el cableado del sistema se corrompe por interferencias o otras causas.
- **Fallo 4: Fallo en la recepción de la señal transmitida por la emisora del "Shadow".**  
No se recibe la señal en el equipo de tierra.
- **Fallo 5: Fallo en el procesamiento de la señal por el hardware del equipo de tierra.**  
La señal es corrompida por el hardware del equipo tierra.
- **Fallo 6: Fallo en el procesamiento de la señal por el software del equipo de tierra.**  
El software del equipo tierra no procesa correctamente la información recibida.

En la figura 1 podemos observar como al producirse cualquier condición de fallo inmediatamente se considerará una pérdida del subsistema.

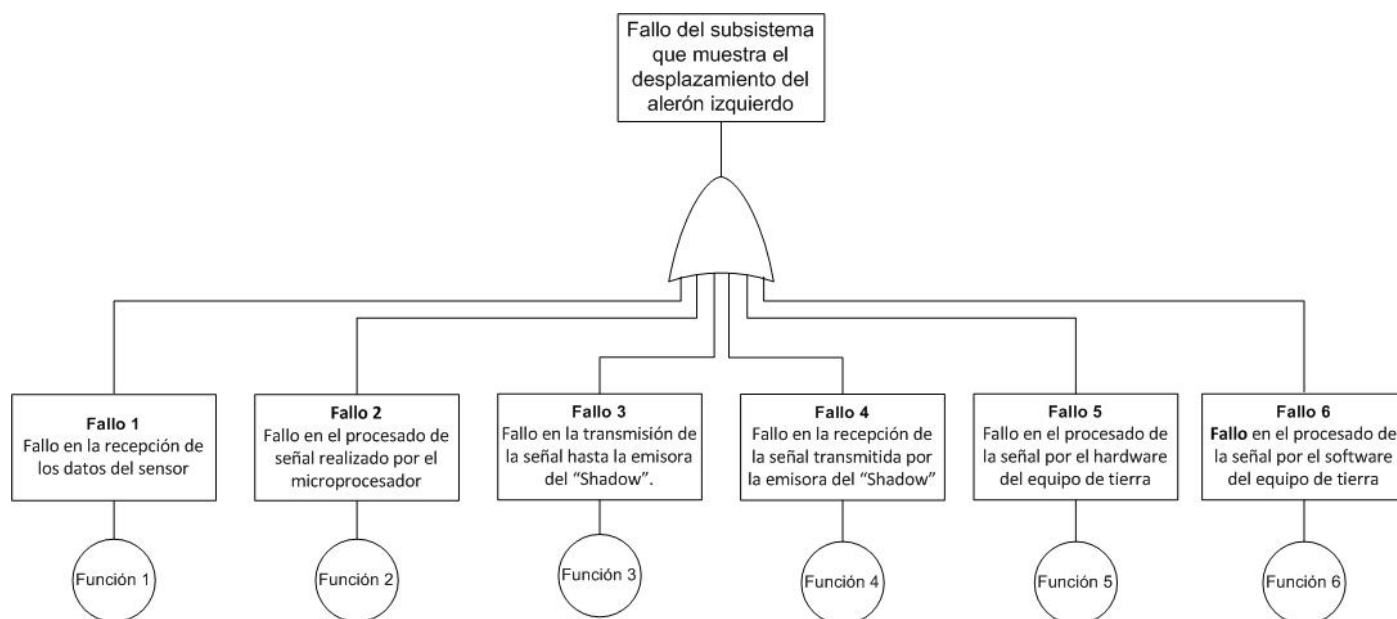


Figura 1: Condiciones de fallo para el alerón izquierdo

En la **Tabla 1** figuran los requisitos que se derivan de estas condiciones de fallo que hemos definido:

<p><b>[RS006] La probabilidad de fallo en la recepción de datos del sensor debe estar clasificada como "Major Failure Condition"</b></p> <p>SW02-02</p> <p>FUENTE: Realización del PSSA.</p> <p>[Diseño arquitectural] [NO IMPLEMENTADO]</p>
<p><b>[RS007] La probabilidad de fallo en el procesado de señal realizado por el microprocesador debe estar clasificada como "Major Failure Condition"</b></p> <p>SW02-02</p> <p>FUENTE: Realización del PSSA.</p> <p>[Diseño arquitectural] [NO IMPLEMENTADO]</p>
<p><b>[RS008] La probabilidad de fallo en la transmisión de la señal hasta la emisora del "Shadow" (Cableado) debe estar clasificada como "Major Failure Condition"</b></p> <p>SW02-02</p> <p>FUENTE: Realización del PSSA.</p> <p>[Diseño arquitectural] [NO IMPLEMENTADO]</p>
<p><b>[RS009] La probabilidad de fallo en la recepción de la señal transmitida por la emisora del "Shadow" debe estar clasificada como "Major Failure Condition"</b></p> <p>SW02-02</p> <p>FUENTE: Realización del PSSA.</p> <p>[Diseño arquitectural] [NO IMPLEMENTADO]</p>
<p><b>[RS010] La probabilidad de fallo en el procesado de la señal por el hardware del equipo de tierra debe estar clasificada como "Major Failure Condition"</b></p> <p>SW02-02</p> <p>FUENTE: Realización del PSSA.</p> <p>[Diseño arquitectural] [NO IMPLEMENTADO]</p>
<p><b>[RS011] La probabilidad de fallo en el procesado de la señal por el software del equipo de tierra debe estar clasificada como "Major Failure Condition"</b></p> <p>SW02-02</p> <p>FUENTE: Realización del PSSA.</p> <p>[Diseño arquitectural] [NO IMPLEMENTADO]</p>

**Tabla 1: Requisitos para las condiciones de fallo del alerón izquierdo.**

Según el documento CS-23, la probabilidad para un "Major Failure Condition" será de 1 fallo cada 100000 horas de vuelo.

#### Condición de fallo del FHA:

*Fallo a la hora de mostrar el desplazamiento angular del alerón derecho.*

Requisitos heredados:

[RS000], [RS002]

Condiciones de fallo:

- Fallo en la recepción de datos del sensor.
- Fallo en el procesado de señal realizado por el microprocesador.
- Fallo en la transmisión de la señal hasta la emisora del "Shadow". Cableado.
- Fallo en la recepción de la señal transmitida por la emisora del "Shadow".
- Fallo en el procesado de la señal por el hardware del equipo de tierra.
- Fallo en el procesado de la señal por el software del equipo de tierra.

En la figura 2 podemos observar como al producirse cualquier condición de fallo inmediatamente se considerará una pérdida del subsistema.

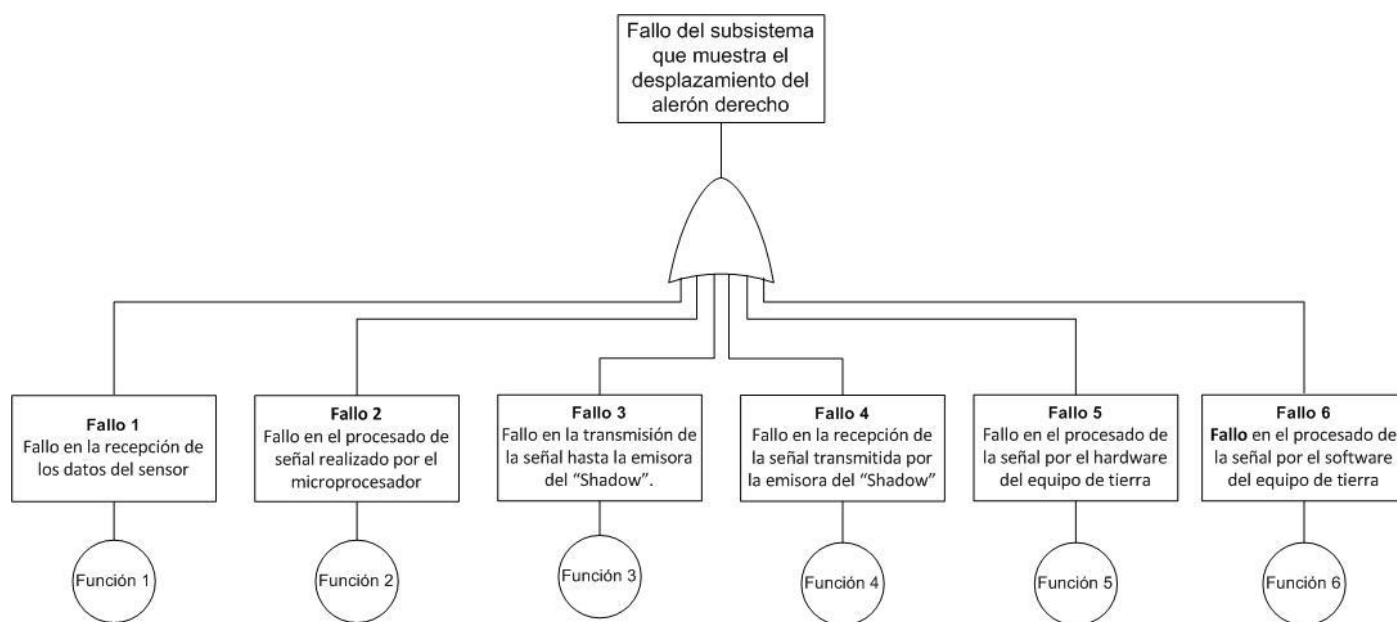


Figura 2: Condiciones de fallo para el alerón derecho



En la **Tabla 2** figuran los requisitos que se derivan de estas condiciones de fallo que hemos definido:

<p><b>[RS012] La probabilidad de fallo en la recepción de datos del sensor debe estar clasificada como "Major Failure Condition"</b></p> <p>SW02-02</p> <p>FUENTE: Realización del PSSA</p> <p>[Diseño arquitectural] [NO IMPLEMENTADO]</p>
<p><b>[RS013] La probabilidad de fallo en el procesamiento de señal realizado por el microprocesador debe estar clasificada como "Major Failure Condition"</b></p> <p>SW02-02</p> <p>FUENTE: Realización del PSSA.</p> <p>[Diseño arquitectural] [NO IMPLEMENTADO]</p>
<p><b>[RS014] La probabilidad de fallo en la transmisión de la señal hasta la emisora del "Shadow" (Cableado) debe estar clasificada como "Major Failure Condition"</b></p> <p>SW02-02</p> <p>FUENTE: Realización del PSSA.</p> <p>[Diseño arquitectural] [NO IMPLEMENTADO]</p>
<p><b>[RS015] La probabilidad de fallo en la recepción de la señal transmitida por la emisora del "Shadow" debe estar clasificada como "Major Failure Condition"</b></p> <p>SW02-02</p> <p>FUENTE: Realización del PSSA.</p> <p>[Diseño arquitectural] [NO IMPLEMENTADO]</p>
<p><b>[RS016] La probabilidad de fallo en el procesamiento de la señal por el hardware del equipo de tierra debe estar clasificada como "Major Failure Condition"</b></p> <p>SW02-02</p> <p>FUENTE: Realización del PSSA.</p> <p>[Diseño arquitectural] [NO IMPLEMENTADO]</p>
<p><b>[RS017] La probabilidad de fallo en el procesamiento de la señal por el software del equipo de tierra debe estar clasificada como "Major Failure Condition"</b></p> <p>SW02-02</p> <p>FUENTE: Realización del PSSA.</p> <p>[Diseño arquitectural] [NO IMPLEMENTADO]</p>

Tabla 2: Requisitos para las condiciones de fallo del alerón derecho

Según el documento CS-23, la probabilidad para un "Major Failure Condition" será de 1 fallo cada 100000 horas de vuelo.

#### Condición de fallo del FHA:

*Fallo a la hora de mostrar el desplazamiento angular del timón de profundidad.*

Requisitos heredados:

[RS000], [RS003]

Condiciones de fallo:

- Fallo en la recepción de datos del sensor.
- Fallo en el procesado de señal realizado por el microprocesador.
- Fallo en la transmisión de la señal hasta la emisora del "Shadow". Cableado.
- Fallo en la recepción de la señal transmitida por la emisora del "Shadow".
- Fallo en el procesado de la señal por el hardware del equipo de tierra.
- Fallo en el procesado de la señal por el software del equipo de tierra.

En la figura 3 podemos observar como al producirse cualquier condición de fallo inmediatamente se considerará una pérdida del subsistema.

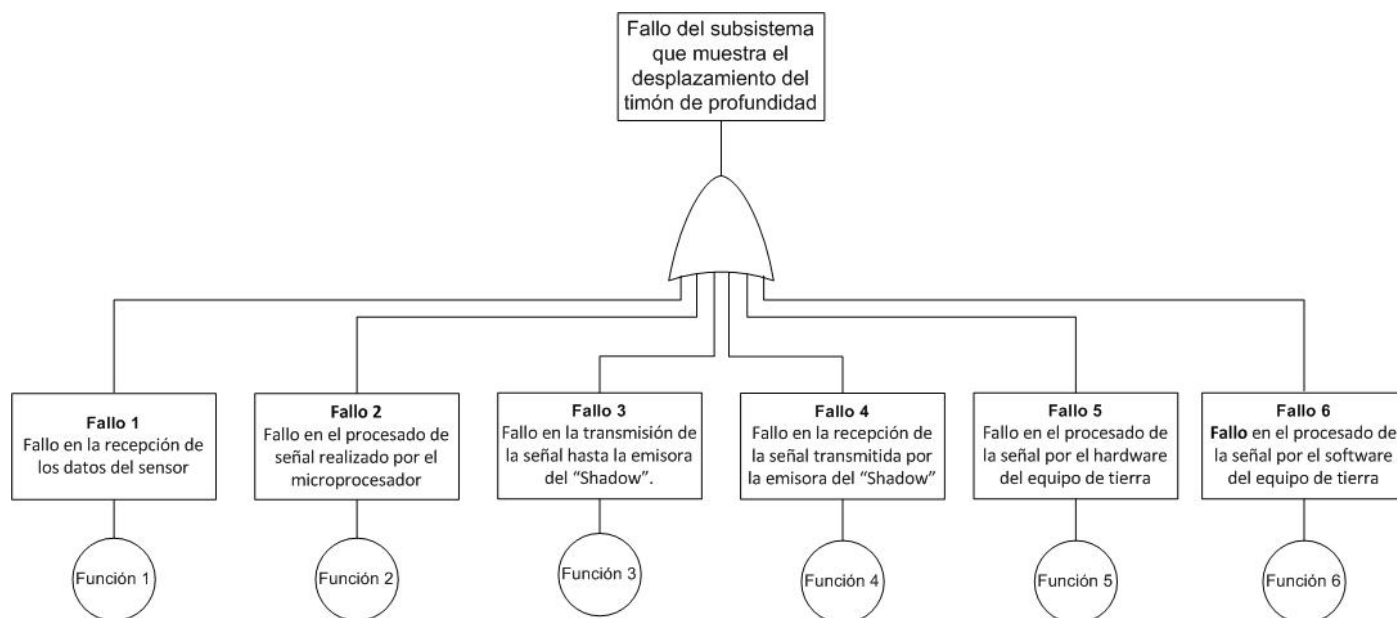


Figura 3: Condiciones de fallo para el timón de profundidad.

En la **Tabla 3** figuran los requisitos que se derivan de estas condiciones de fallo que hemos definido:

<p><b>[RS018] La probabilidad de fallo en la recepción de datos del sensor debe estar clasificada como "Major Failure Condition"</b></p> <p>SW02-02</p> <p>FUENTE: Realización del PSSA</p> <p>[Diseño arquitectural] [NO IMPLEMENTADO]</p>
<p><b>[RS019] La probabilidad de fallo en el procesado de señal realizado por el microprocesador debe estar clasificada como "Major Failure Condition"</b></p> <p>SW02-02</p> <p>FUENTE: Realización del PSSA.</p> <p>[Diseño arquitectural] [NO IMPLEMENTADO]</p>
<p><b>[RS020] La probabilidad de fallo en la transmisión de la señal hasta la emisora del "Shadow" (Cableado) debe estar clasificada como "Major Failure Condition"</b></p> <p>SW02-02</p> <p>FUENTE: Realización del PSSA.</p> <p>[Diseño arquitectural] [NO IMPLEMENTADO]</p>
<p><b>[RS021] La probabilidad de fallo en la recepción de la señal transmitida por la emisora del "Shadow" debe estar clasificada como "Major Failure Condition"</b></p> <p>SW02-02</p> <p>FUENTE: Realización del PSSA.</p> <p>[Diseño arquitectural] [NO IMPLEMENTADO]</p>
<p><b>[RS022] La probabilidad de fallo en el procesado de la señal por el hardware del equipo de tierra debe estar clasificada como "Major Failure Condition"</b></p> <p>SW02-02</p> <p>FUENTE: Realización del PSSA.</p> <p>[Diseño arquitectural] [NO IMPLEMENTADO]</p>
<p><b>[RS023] La probabilidad de fallo en el procesado de la señal por el software del equipo de tierra debe estar clasificada como "Major Failure Condition"</b></p> <p>SW02-02</p> <p>FUENTE: Realización del PSSA.</p> <p>[Diseño arquitectural] [NO IMPLEMENTADO]</p>

**Tabla 3: Requisitos para las condiciones de fallo del timón de profundidad**

Según el documento CS-23, la probabilidad para un "Major Failure Condition" será de 1 fallo cada 100000 horas de vuelo.

#### Condición de fallo del FHA:

*Fallo a la hora de mostrar el desplazamiento angular del timón de dirección.*

Requisitos heredaros:

[RS000], [RS004]

Condiciones de fallo:

- Fallo en la recepción de datos del sensor.
- Fallo en el procesado de señal realizado por el microprocesador.
- Fallo en la transmisión de la señal hasta la emisora del "Shadow". Cableado.
- Fallo en la recepción de la señal transmitida por la emisora del "Shadow".
- Fallo en el procesado de la señal por el hardware del equipo de tierra.
- Fallo en el procesado de la señal por el software del equipo de tierra.

En la figura 4 podemos observar como al producirse cualquier condición de fallo inmediatamente se considerará una pérdida del subsistema.

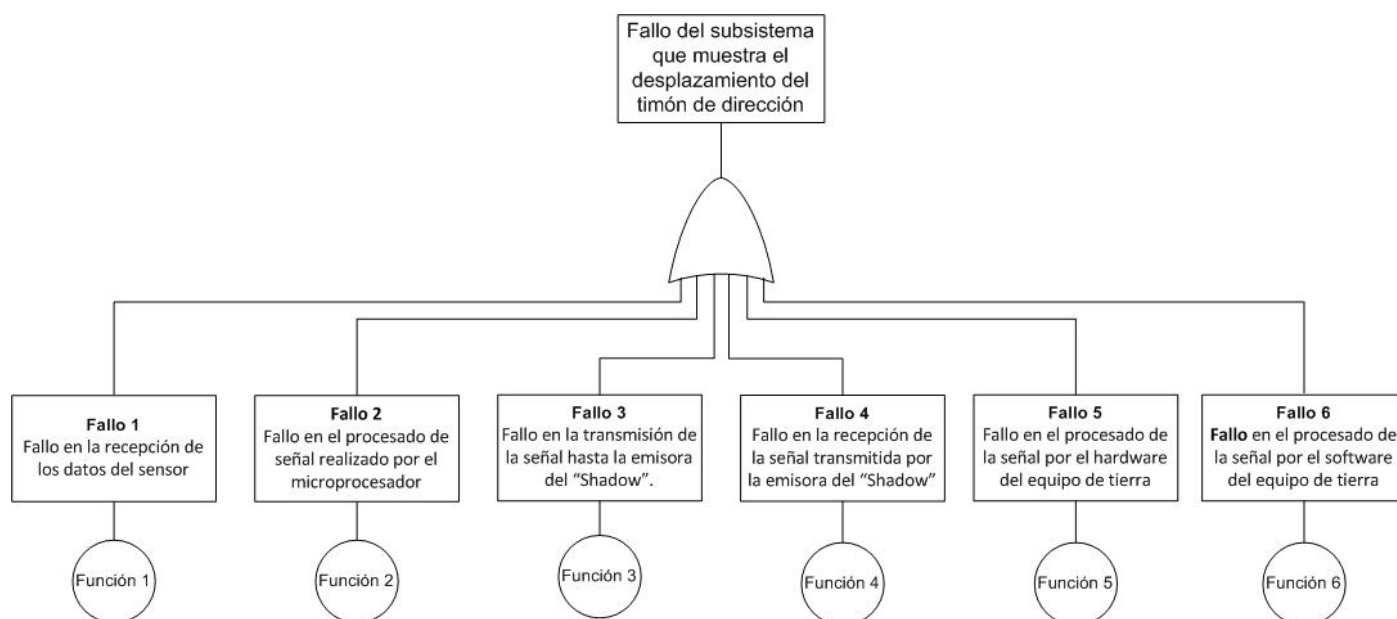


Figura 4: Condiciones de fallo para el timón de dirección.

En la **Tabla 4** figuran los requisitos que se derivan de estas condiciones de fallo que hemos definido:

<p><b>[RS024] La probabilidad de fallo en la recepción de datos del sensor debe estar clasificada como "Major Failure Condition"</b></p> <p>SW02-02</p> <p>FUENTE: Realización del PSSA</p> <p>[Diseño arquitectural] [NO IMPLEMENTADO]</p>
<p><b>[RS025] La probabilidad de fallo en el procesamiento de señal realizado por el microprocesador debe estar clasificada como "Major Failure Condition"</b></p> <p>SW02-02</p> <p>FUENTE: Realización del PSSA.</p> <p>[Diseño arquitectural] [NO IMPLEMENTADO]</p>
<p><b>[RS026] La probabilidad de fallo en la transmisión de la señal hasta la emisora del "Shadow" (Cableado) debe estar clasificada como "Major Failure Condition"</b></p> <p>SW02-02</p> <p>FUENTE: Realización del PSSA.</p> <p>[Diseño arquitectural] [NO IMPLEMENTADO]</p>
<p><b>[RS027] La probabilidad de fallo en la recepción de la señal transmitida por la emisora del "Shadow" debe estar clasificada como "Major Failure Condition"</b></p> <p>SW02-02</p> <p>FUENTE: Realización del PSSA.</p> <p>[Diseño arquitectural] [NO IMPLEMENTADO]</p>
<p><b>[RS028] La probabilidad de fallo en el procesamiento de la señal por el hardware del equipo de tierra debe estar clasificada como "Major Failure Condition"</b></p> <p>SW02-02</p> <p>FUENTE: Realización del PSSA.</p> <p>[Diseño arquitectural] [NO IMPLEMENTADO]</p>
<p><b>[RS029] La probabilidad de fallo en el procesamiento de la señal por el software del equipo de tierra debe estar clasificada como "Major Failure Condition"</b></p> <p>SW02-02</p> <p>FUENTE: Realización del PSSA.</p> <p>[Diseño arquitectural] [NO IMPLEMENTADO]</p>

Tabla 4: Requisitos para las condiciones de fallo del timón de dirección.

Según el documento CS-23, la probabilidad para un "Major Failure Condition" será de 1 fallo cada 100000 horas de vuelo.

## 4 EVALUAR LAS DECISIONES DE DISEÑO CON RESPECTO A LOS REQUISITOS Y OBJETIVOS DEL SISTEMA

Destacar que, como muestra la **Figura 5** extraída del FHA, cualquier fallo en uno de los subsistemas que conforman nuestro sistema conduce ya directamente a la pérdida del sistema.

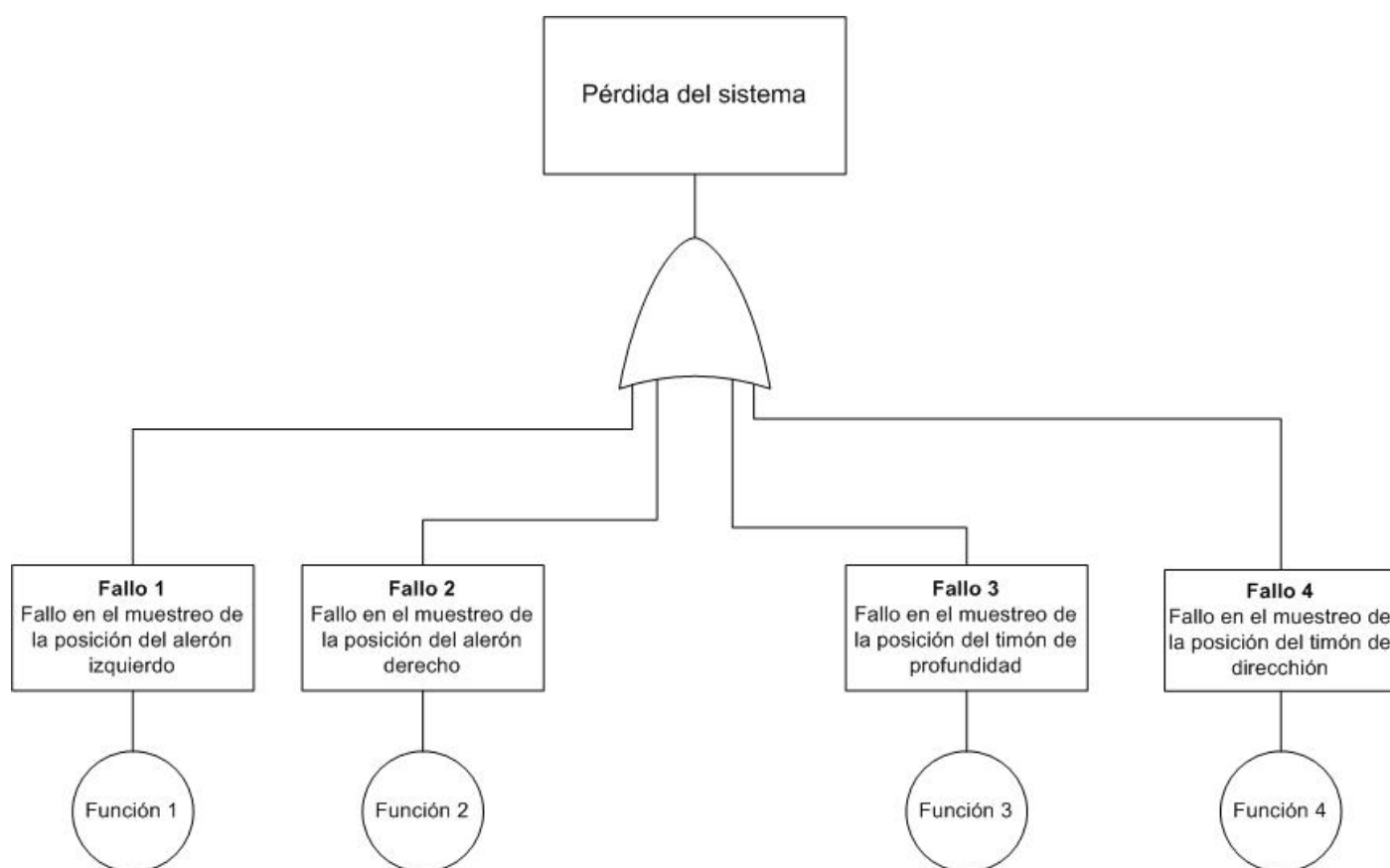


Figura 5: Diagrama de pérdida del sistema

Según esta recogido en el documento ARP4761, en este apartado deberemos mostrar, utilizando un FTA o algún método similar, que los requisitos y objetivos asociados a las condiciones de fallo pueden ser demostrados con la arquitectura propuesta.

Como se observa en la **Figura 6** extraída del documento de diseño arquitectural, nuestra arquitectura se divide en cuatro subsistemas completamente independientes entre ellos enlazados a dos subsistemas globales. Con esto buscamos asegurar que las probabilidades de fallo no se relacionan entre sí.

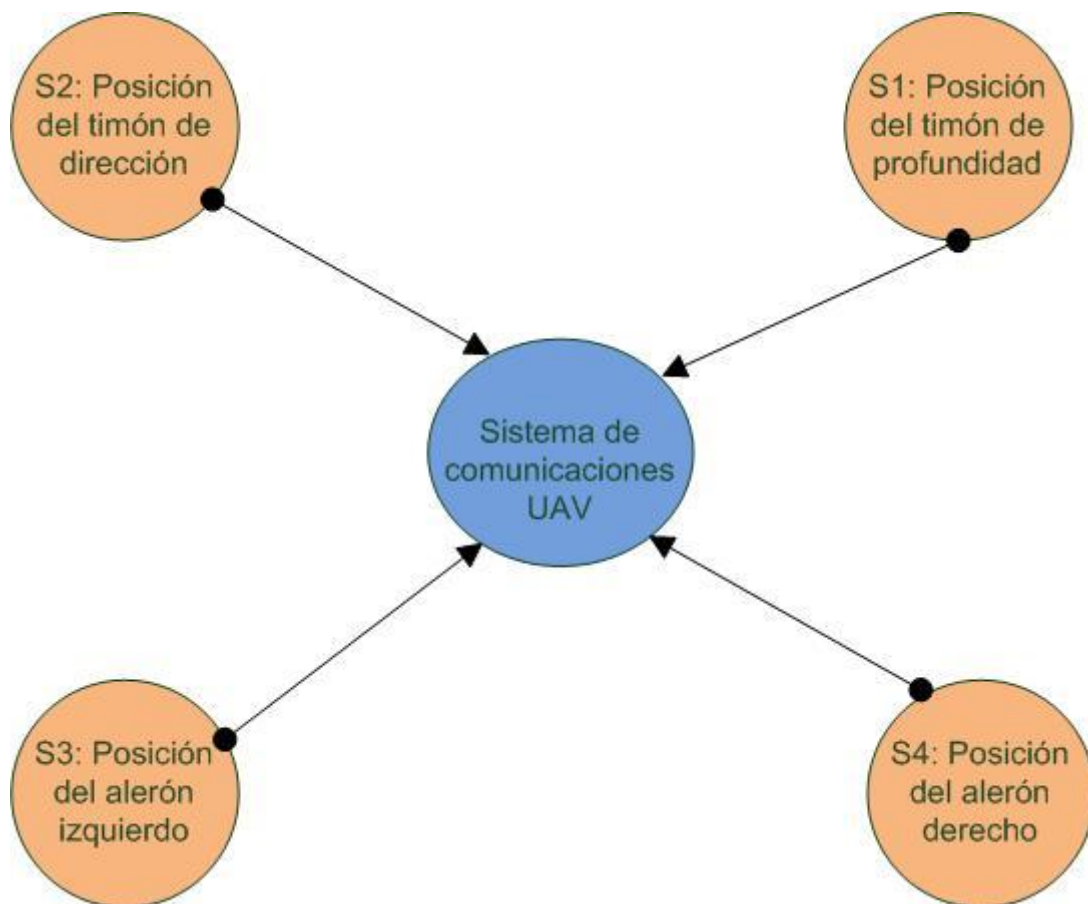


Figura 6: Arquitectura del sistema

Según el documento "SAE ARP4761" en este apartado también debemos mencionar las tareas de mantenimiento asociadas a nuestro sistema que puedan ayudar en medida de lo posible a satisfacer los requisitos de Safety. Según este mismo documento también será necesario definir un Developement Assurance Level (DAL) para todos nuestros sistemas.

#### Tareas de mantenimiento:

Tanto el hardware como el software que forman parte del sistema carecen de defectos de degradación temporal, por esto es innecesario imputar unas obligaciones de mantenimiento para cumplir con los requisitos de safety.

#### DAL de los diferentes sistemas:

Según la **Tabla 5** extraída del documento "SAE ARP4754" capítulo 3, podemos observar como para una "Major Failure Condition" le corresponde un DAL C.

Failure Condition Classification	System Development Assurance Level
Catastrophic	A
Hazardous/Severe Major	B
Major	C
Minor	D
No Safety Effect	E

Tabla 5: Clasificación de las condiciones de fallo según su DAL.

Por este motivo, podemos ver como en la **Figura 7** a todos nuestros sistemas les corresponde un DAL C:

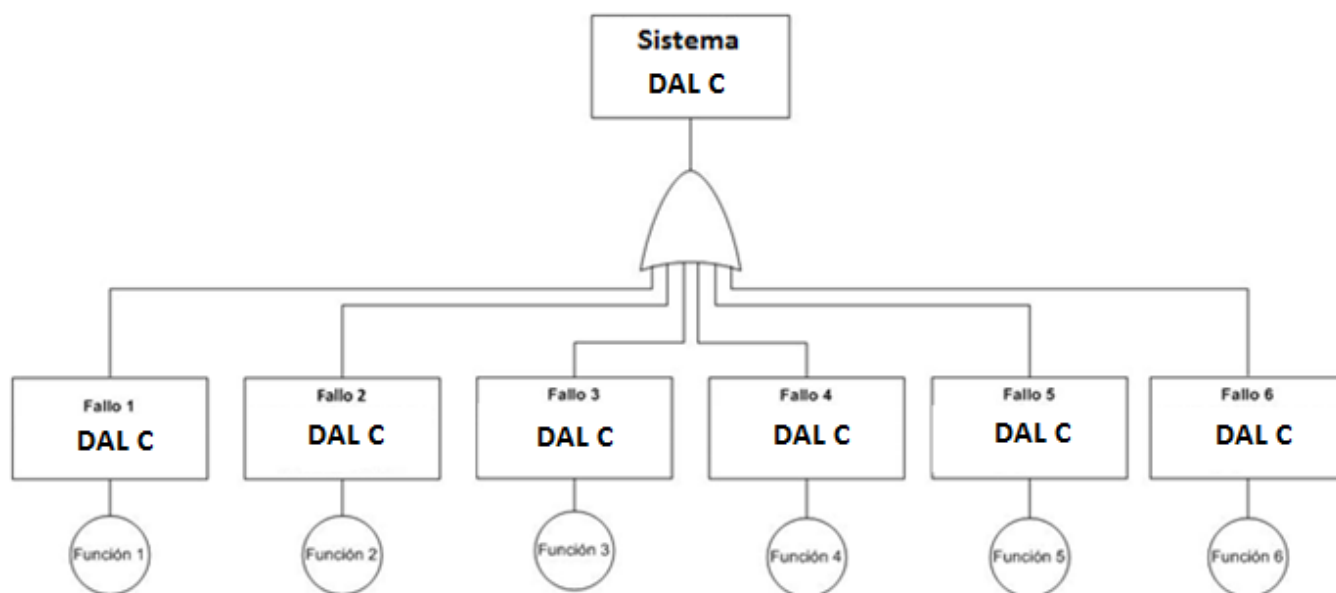


Figura 7: DAL de los diferentes subsistemas



## 5 DERIVAR LOS REQUISITOS DE “SAFETY” AL DISEÑO DE LOS SUBSISTEMAS

Según esta recogido en el documento ARP4761, en este punto deberemos enlazar nuestros requisitos a directrices que el departamento de diseño tendrá que tener en cuenta al elegir el hardware y software que formará parte de nuestro equipo.

### 5.1 LISTA DE CONDICIONES DE FALLO ACTUALIZADA QUE INCLUYA UNAS DIRECTRICES DE CÓMO LOS REQUISITOS PUEDEN SER SATISFECHOS.

Según esta recogido en el documento ARP4761, para este punto elaboraremos una tabla donde figuren por este orden:

- Condición de fallo
- Requisito o requisitos derivados de la condición de fallo.
- Directrices de cómo estos requisitos pueden darse por satisfechos o no.

Fallo en la recepción de datos del sensor.	[RS006]	-Comprobar que la probabilidad de fallo del sistema es inferior a una cada 10.000 horas de vuelo. Para esto nos ayudaremos de un banco de pruebas y un sistema de ciclos acelerados.
	[RS012]	
	[RS018]	
	[RS024]	
		-Consideraremos posible fallo del potenciómetro cuando:
		-Información recibida incorrecta (25%).
		-No recibir información.
		-Recibir código de seguridad.

Fallo en el procesado de señal realizado por el microprocesador.	[RS007]	-Comprobar que la probabilidad de fallo del sistema es inferior a una cada 10.000 horas de vuelo. Para esto nos ayudaremos de un banco de pruebas y un sistema de ciclos acelerados.
	[RS013]	
	[RS019]	
	[RS025]	
		-Consideraremos posible fallo del microprocesador cuando:
		-Información recibida incorrecta(25%).
		-No recibir información.
		-Recibir código de seguridad (50%).

Fallo en la transmisión de la señal hasta la emisora del “Shadow”. Cableado.	[RS008]	-Comprobar que la probabilidad de fallo del sistema es inferior a una cada 10.000 horas de vuelo. Para esto nos ayudaremos de un banco de pruebas y un sistema de ciclos acelerados.
	[RS014]	
	[RS020]	

[RS026]

- Consideraremos posible fallo del cableado cuando:
  - No recibir información.
  - Recibir código de seguridad (50%).

Fallo en la recepción de la señal transmitida por la emisora del "Shadow".

[RS009]

[RS015]

[RS021]

[RS027]

- Comprobar que la probabilidad de fallo del sistema es inferior a una cada 10.000 horas de vuelo. Para esto nos ayudaremos de un banco de pruebas y un sistema de ciclos acelerados.
- Consideraremos posible fallo del Wireless cuando:
  - No recibir información(100%).

Fallo en el procesado de la señal por el hardware del equipo de tierra.

[RS010]

[RS016]

[RS022]

[RS028]

- Comprobar que la probabilidad de fallo del sistema es inferior a una cada 10.000 horas de vuelo. Para esto nos ayudaremos de un banco de pruebas y un sistema de ciclos acelerados.
- Consideraremos fallo del hardware del sistema tierra cuando la información recibida por el receptor no llegue al software.

Fallo en el procesado de la señal por el software del equipo de tierra.

[RS011]

[RS017]

[RS023]

[RS029]

- Comprobar que la probabilidad de fallo del sistema es inferior a una cada 10.000 horas de vuelo. Para esto nos ayudaremos de un banco de pruebas y un sistema de ciclos acelerados.
- Consideraremos fallo del software del sistema tierra cuando los datos recibidos no se muestren de forma correcta al cliente.

**Tabla 6:** Condiciones de fallo actualizadas

## 5.2 LISTA CON LOS REQUISITOS DE "SAFETY" INTERRELACIONADOS CON EL HARDWARE Y SOFTWARE.

Según esta recogido en el documento ARP4761, En este punto valoraremos que partes del equipo intervienen o no en el cumplimiento de un determinado requisito. Mostraremos esta interrelación con una tabla que constará de la siguiente estructura:

- Condición de fallo.
- Requisito o requisitos derivados de esa condición de fallo.
- Software/hardware interrelacionado.

Fallo en la recepción de datos del sensor.	[RS006]	Potenciómetro
	[RS012]	
	[RS018]	
	[RS024]	
Fallo en el procesado de señal realizado por el microprocesador.	[RS007]	microprocesador
	[RS013]	
	[RS019]	
	[RS025]	
Fallo en la transmisión de la señal hasta la emisora del "Shadow". Cableado.	[RS008]	Cableado
	[RS014]	
	[RS020]	
	[RS026]	
Fallo en la recepción de la señal transmitida por la emisora del "Shadow".	[RS009]	Emisor
	[RS015]	Receptor
	[RS021]	
	[RS027]	
Fallo en el procesado de la señal por el hardware del equipo de tierra.	[RS010]	Hardware equipo tierra
	[RS016]	
	[RS022]	
	[RS028]	
Fallo en el procesado de la señal por el software del equipo de tierra.	[RS011]	Software equipo tierra
	[RS017]	
	[RS023]	
	[RS029]	

**Tabla 7:** Requisitos de "safety" interrelacionados con el hardware y software

### 5.3 REQUISITOS PARA LA INSTALACIÓN DEL SISTEMA

Según esta recogido en el documento ARP4761, en este punto deberemos estudiar todos aquellos parámetros o implicaciones de la instalación de nuestro sistema en la aeronave. Se trata de elaborar un listado de requisitos que remitiremos al equipo de diseño. En nuestro caso, los requisitos que hemos creído oportuno reportar al departamento de diseño están recogidos en la **tabla 8**.

**[RS035] La señal no se verá afectada por el ruido generado en el avión.**

SW02-02

FUENTE: Realización del PSSA

[Diseño arquitectural] [NO IMPLEMENTADO]

**[RS036] El sistema no afectará al correcto comportamiento mecánico de las superficies de control**

SW02-02

FUENTE: Realización del PSSA.

[Diseño arquitectural] [NO IMPLEMENTADO]

**[RS037] El sistema no interferirá en la aerodinámica de la aeronave**

SW02-02

FUENTE: Realización del PSSA.

[Diseño arquitectural] [NO IMPLEMENTADO]

Tabla 8: Requisitos para la instalación del sistema.

## 5.4 DAL DEL SOFTWARE Y EL HARDWARE

Según la **Tabla 9** extraída del documento "SAE ARP4754" capítulo 3, podemos observar como para una "Major Failure Condition" se corresponde un DAL C.

Failure Condition Classification	System Development Assurance Level
Catastrophic	A
Hazardous/Severe Major	B
Major	C
Minor	D
No Safety Effect	E

Tabla 9: Clasificación de las condiciones de fallo según su DAL.

Entonces:

Hardware	DAL
Potenciómetro	C
microprocesador	C
Cableado	C
Emisor/Receptor	C



Hardware equipo tierra	C
Software equipo tierra	C

Tabla 10: DAL de Software y hardware

## 5.5 TAREAS DE MANTENIMIENTO

Tanto el hardware como el software que forman parte del sistema carecen de defectos de degradación temporal, por esto es innecesario imputar unas obligaciones de mantenimiento para cumplir con los requisitos de safety.

A más no contamos con MTBF de ningún componente del hardware como para aconsejar ninguna de estas tareas.



## 6 BIBLIOGRAFÍA

- SAE ARP4761, "Guidelines and methods for conducting the safety assessment process on civil airborne systems and equipment". U.S.A., 1996



## 7 GLOSARIO

FHA: Functional Hazard Assessment

PSSA: Preliminary System Safety Assessment

ZSA: Zonal Safety Assessment

FTA: Fault Tree Analysis



# Diseño de Sistema Sensor Para superficies de UAV

---

Iss./Rev.: 12/07/2009

Date: 07/07/2009

[END OF DOCUMENT]



# ANEXO 9: Zonal Safety Analysis (ZSA)

	Name	Signature	Date
Proyectista 1	Eloy Herrera		12/07/09
Proyectista 2	Víctor de los Santos		12/07/09
Director	Jorge Ramírez Alcantara		12/07/09

## Registro de cambios

Fecha	Cambio
10/07/09	-Se cambia la introducción por otra que ofrece una guía más detallada de cómo elaborar un ZSA
12/07/09	<ul style="list-style-type: none"><li>- Se cambia el template.</li><li>- Se añaden referencias al diseño.</li><li>- Se modifican requisitos erróneos.</li><li>- Se añaden ilustraciones.</li></ul>

<b>1</b>	<b>INTRODUCCIÓN .....</b>	<b>6</b>
<b>2</b>	<b>LOCALIZACIÓN DE NUESTRO SISTEMA DENTRO DE LA AERONAVE.....</b>	<b>9</b>
<b>3</b>	<b>IDENTIFICACIÓN DE TODOS LOS SUBSISTEMAS QUE SEAN UN PROBLEMA EN POTENCIA....</b>	<b>14</b>
<b>4</b>	<b>COMO EVITAREMOS LOS ANTERIORES PROBLEMAS POTENCIALES.....</b>	<b>16</b>
<b>5</b>	<b>BIBLIOGRAFÍA .....</b>	<b>18</b>
<b>6</b>	<b>GLOSARIO .....</b>	<b>19</b>

TABLA 1: REQUISITOS DERIVADOS DE LA ELABORACIÓN DEL ZSA. ....	15
TABLA 2: PROPIEDADES DEL POLICLORURO DE VINILO (PVC) .....	17

## Índice de Figuras

FIGURA 1: EJEMPLO DE DIVISIÓN DE UNA AERONAVE .....	6
FIGURA 2: TRAZADO DE LAS LINIAS DE COMUNICACIÓN EN EL "SHADOW" .....	9
FIGURA 3: SERVO DEL ALERÓN IZQUIERDO VISTO DE PERFIL .....	10
FIGURA 4: SERVO DEL ALERÓN DERECHO VISTO EN PLANTA.....	10
FIGURA 5: PLANTA DE LA CAVIDAD DEL SERVO DEL ALERÓN .....	10
FIGURA 6: ALZADO DE LA CAVIDAD DEL SERVO DEL ALERÓN.....	10
FIGURA 7: SERVO DEL TIMÓN DE DIRECCIÓN .....	11
FIGURA 8: TUBERÍA QUE UNE EL ESTABILIZADOR DE COLA CON EL CUERPO DE LA AERONAVE .....	11
FIGURA 9: SERVOS DEL TIMÓN DE PROFUNDIDAD VISTOS DE PERFIL.....	12
FIGURA 10: SERVOS DEL TIMÓN DE PROFUNDIDAD VISTOS EN PLANTA.....	12
FIGURA 11: PLANTA DE LA CAVIDAD DEL SERVO DEL TIMÓN DE PROFUNDIDAD.....	13
FIGURA 12: ALZADO DE LA CAVIDAD DEL SERVO DEL TIMÓN DE PROFUNDIDAD.....	13
FIGURA 13: NIVELES PRIMARIOS UN UNA AERONAVE. ....	14
FIGURA 14: CABLEADO A EXTRAER DEL SERVO.....	16
FIGURA 15: DIMENSIONES DE LA CAJA QUE CONTENDRÁ EL SISTEMA .....	17

## 1 INTRODUCCIÓN

Tomando como referencia el documento de diseño arquitectural se hizo una inspección de aquellas zonas que alojarían nuestro sistema y de cómo la implementación de nuestro sistema en la aeronave podría interferir o alterar su funcionamiento original. Teniendo en cuenta el consumo del hardware, los problemas que se podrían derivar de embarcar el sistema en el UAV y el posible funcionamiento erróneo de nuestro sistema.

Muchas veces los análisis de safety caen en el error de basarse en esquemas rígidos. Estos esquemas muchas veces no tienen en consideración las implicaciones de instalar físicamente el sistema en una aeronave. Debemos tener en cuenta que los diferentes subsistemas que forman parte del avión no son independientes entre ellos. Por ello es necesario hacer una análisis que tenga en cuenta las implicaciones que la instalación de un sistema puntual pueda tener en los sistemas próximos a él. Éste análisis es llamado zonal safety analysis.

En un ZSA debemos analizar zona por zona de la aeronave. La división del avión por partes, por tanto, será una tarea previa a la realización de este análisis. Podemos consultar un ejemplo de división en la **Figura 1**.

Los resultados extraídos de este análisis serán entradas importantes en la elaboración del SSA y complementarán a sus requisitos.

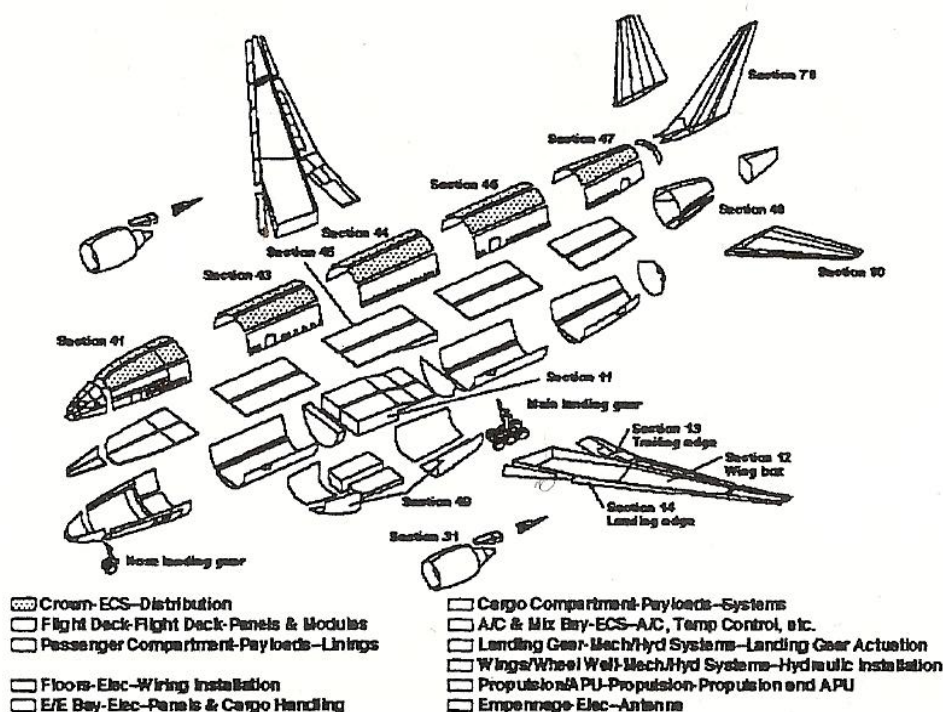


Figura 1: Ejemplo de división de una aeronave

El objetivo de un Zonal Safety Analysis es asegurar que el diseño del sistema y su instalación cumple con los estándares básicos de diseño y instalación, el efecto de los fallos del sistema en la aeronave, la implicación de los errores de mantenimiento y que cumple con los requisitos de independencia especificados en el FTA.

Durante el proceso de ZSA deberemos evaluar todas las zonas dónde está previsto instalar nuestro sistema, describiendo que encontramos en cada zona y los espacios libres dónde ubicaremos nuestros sistemas. Una vez identificados todos los elementos



## Diseño de Sistema Sensor Para superficies de UAV

Iss./Rev.: 12/07/2009

Date: 08/07/2009

---

cercanos deberemos hacer una lista con cada uno de los sistemas que pueden derivar en un malfuncionamiento del sistema y explicar que efectos tendría en el comportamiento de la aeronave.

Los resultados de éste ZSA estarán incluidos en el reporte, y en caso de que exista alguna desviación frente al diseño inicial, ésta se reportará al departamento de diseño dónde deberán efectuar los cambios pertinentes.



## 2 OBJETIVO

Realizar el Zonal Safety Assessment (ZSA) del sistema sensor que estamos diseñando.



### 3 LOCALIZACIÓN DE NUESTRO SISTEMA DENTRO DE LA AERONAVE

Nuestro sistema está pensado para implantarse en todas las superficies de control de cualquier aeronave. En este caso nos centraremos en la plataforma "Shadow" donde colocaremos un sensor en cada uno de los servos de los dos alerones, uno en el timón de profundidad y otro en el timón de dirección. El cableado desde nuestros sistemas a la estación emisora del Shadow se pasará por los mismos conductos que utiliza el cableado del sistema de control. En la **Figura 2**, proporcionada por el departamento de diseño, podemos observar el recorrido del cableado de control.

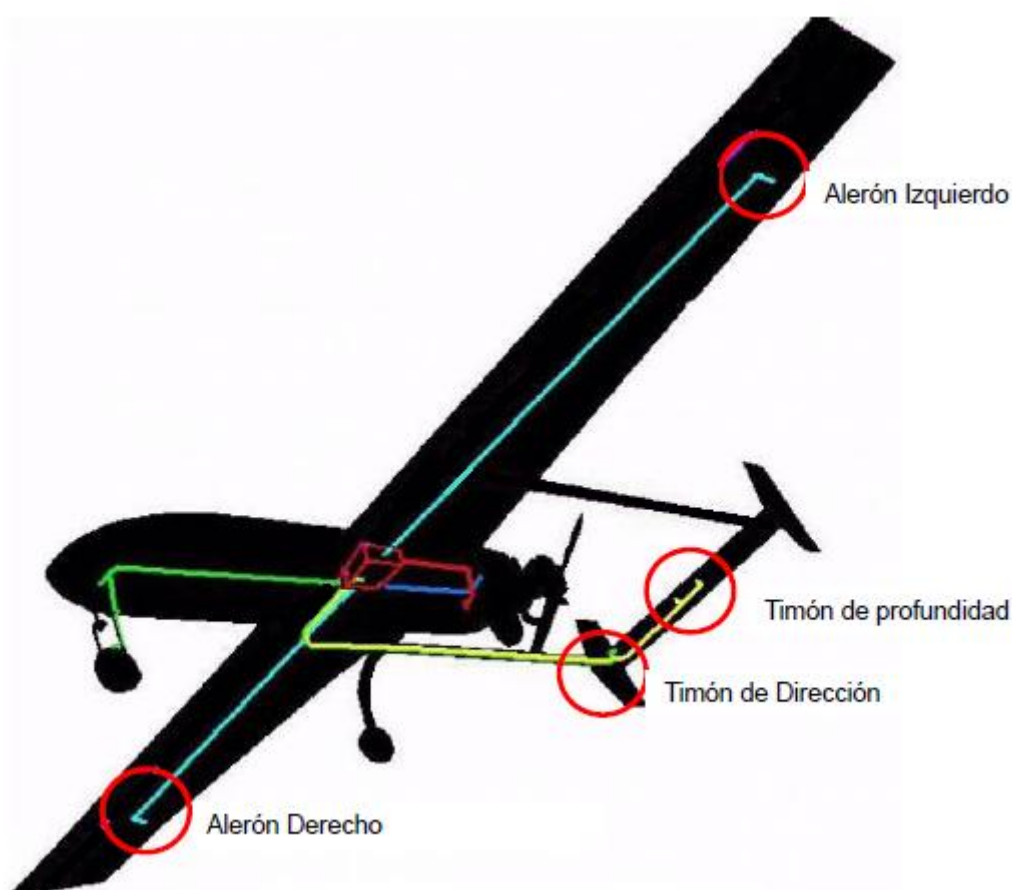


Figura 2: Trazado de las líneas de comunicación en el "Shadow".



## Alerones

Según la última revisión del documento de diseño el sensor de los alerones se alojará en el interior del ala ya que ésta está completamente hueca como se observa en la **Figura 3** y la **Figura 6**. En la zona escogida por el departamento de diseño encontraremos a más del servomotor del que tomaremos la medida, el cableado de alimentación de éste y las líneas de comunicación del sistema de control. El espacio concreto con el que contamos se describe en la **Figura 5** y la **Figura 6**.

Desde el departamento de safety recomendamos el instalar nuestro sistema lo más lejos posible de las líneas de control y buscar métodos para asegurar la correcta fijación del sistema a la superficie del ala.



Figura 3: Servo del alerón izquierdo visto de perfil.



Figura 4: Servo del alerón derecho visto en planta.

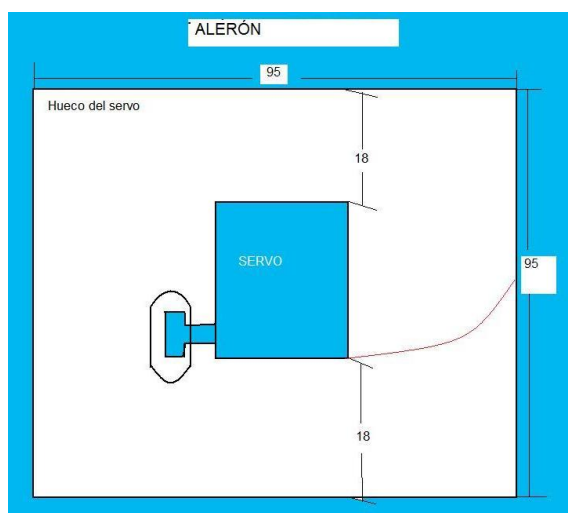


Figura 5: Planta de la cavidad del servo del alerón

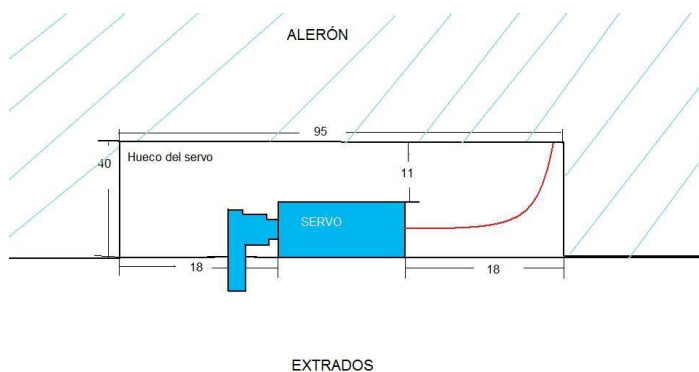


Figura 6: Alzado de la cavidad del servo del alerón



## Timón de dirección

Según la última revisión del documento de diseño el sensor de los alerones se alojará en el interior de la tubería que une las alas con las superficies de control de guiñada y picado. La zona escogida es una tubería hueca por la que pasan los conductos de control y el cableado de alimentación de los servomotores como podemos observar en la **Figura 8**. El espacio para la colocación de nuestro sistema se asume como suficiente.

Podemos observar una fotografía del alerón en la **Figura 7**.

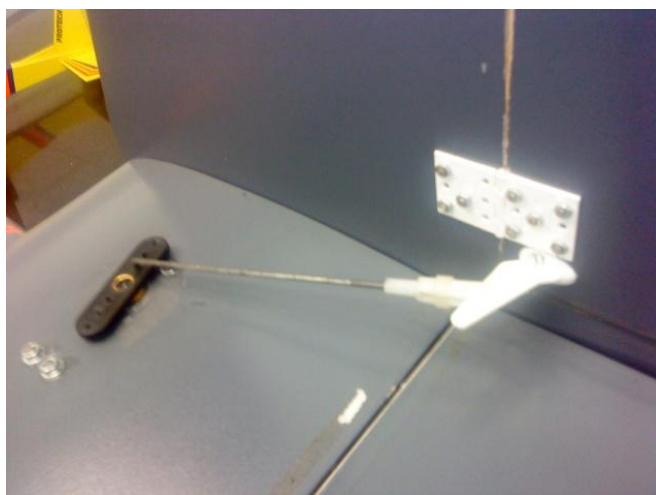


Figura 7: Servo del timón de dirección.



Figura 8: Tubería que une el estabilizador de cola con el cuerpo de la aeronave.



## Timón de profundidad

Según la última revisión del documento de diseño el sensor de los alerones se alojará en el interior del estabilizador horizontal de cola ya que ésta está completamente hueca. En la zona escogida por el departamento de diseño encontraremos a más del servomotor del que tomaremos la medida, el cableado de alimentación de éste y las líneas de comunicación del sistema de control. Tenemos las dimensiones del hueco en la **Figura 11** y la **Figura 12**.

Desde el departamento de safety recomendamos el instalar nuestro sistema lo más lejos posible de las líneas de control y buscar métodos para asegurar la correcta fijación del sistema a la superficie del ala.

En la **Figura 9** y la **Figura 10** podemos observar unas fotografías de los servos que mueven esta superficie.



Figura 9: Servos del timón de profundidad vistos de perfil.



Figura 10: Servos del timón de profundidad vistos en planta.

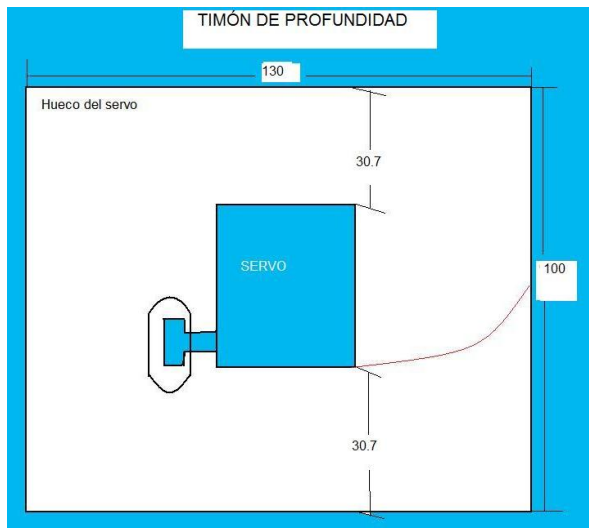


Figura 11: Planta de la cavidad del servo del timón de profundidad

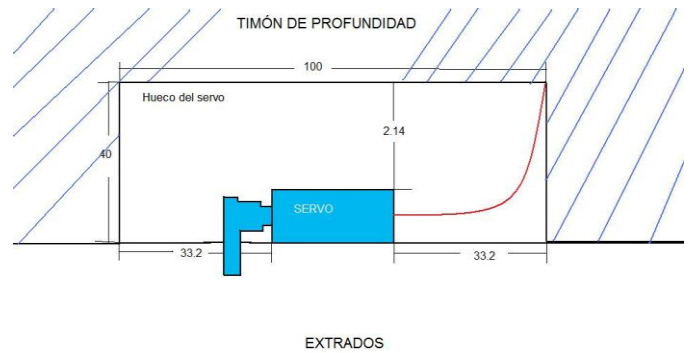


Figura 12: Alzado de la cavidad del servo del timón de profundidad

## 4 IDENTIFICACIÓN DE TODOS LOS SUBSISTEMAS QUE SEAN UN PROBLEMA EN POTENCIA.

### a) Potenciómetro

Al soldar dos cables al potenciómetro para tomar una medida de la resistencia de éste, estamos tocando un subsistema perteneciente a la función de control de la aeronave, que como podemos ver en la **Figura 13** es crítica para el funcionamiento del avión. Por lo que según el documento NASA/TM-2007-214539 nuestro sistema pasa de tener una “major condition failure” a “catastrophic condition failure”. Esto ahora implica que todas aquellas funciones de nuestro sistema (y todos los subsistemas que forman parte de ellas) que puedan interferir en el correcto funcionamiento del potenciómetro pasan a tener el mismo DAL que el servomotor. En nuestro caso se trata de un DAL A al no haber servomotores redundantes. Por este motivo los requisitos de safety [RS000], [RS030] y [RS031] se actualizarán para cumplir con esta nueva condición. La máxima prioridad ahora será no falsear la medida que el sistema de control propio del servomotor toma del potenciómetro. Reportaremos a diseño los requisitos derivados: [RS030], [RS031].

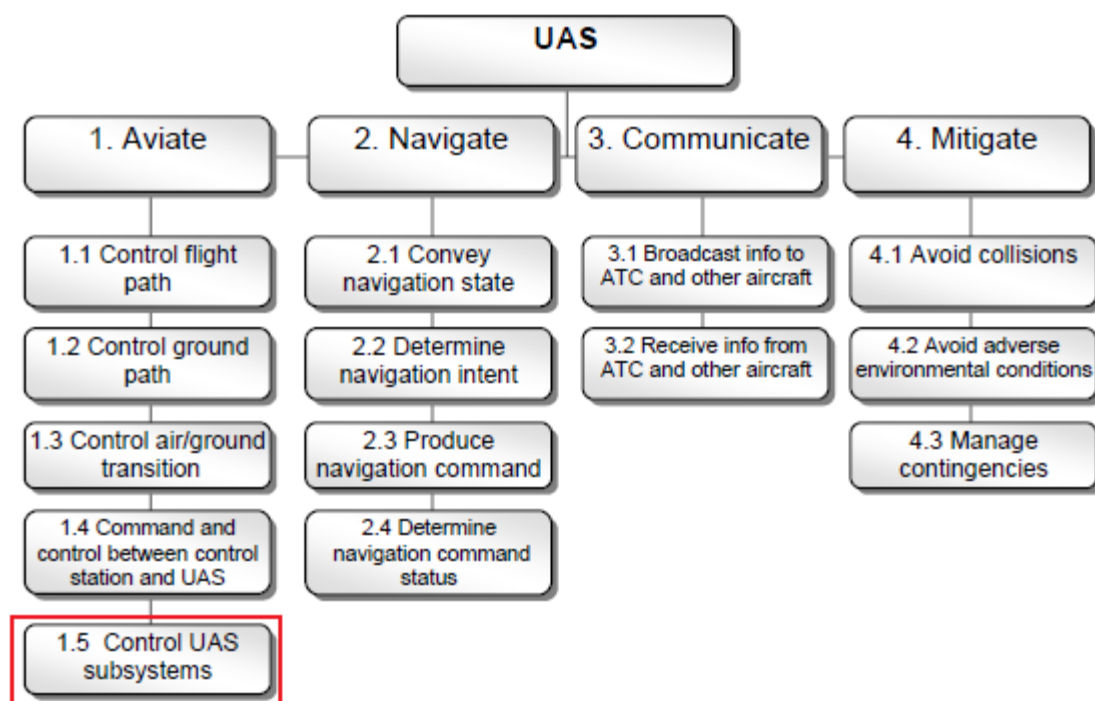


Figura 13: Niveles primarios en una aeronave.

### b) Encapsulado del servomotor.

Tal y como figura en la guía de instalación deberemos hacer pasar cableado desde el interior del servo a su exterior, por esto deberemos perforar el encapsulado. Esto podría tener consecuencias en el aislamiento del servo (temperatura, agentes climáticos, pequeña fauna, etc.), comprometiendo así su seguridad. Reportaremos a diseño los requisitos derivados: [RS032]

### c) Encapsulado de nuestro sistema

El encapsulado de nuestro sistema tendrá que adecuarse a los espacios que nos brinda la arquitectura del UAV y primando en todo momento que éste no interfiera con los sistemas mecánicos de la aeronave. Los detalles de estos espacios están

explicados en el capítulo 2 de este mismo documento. Reportaremos a diseño los requisitos derivados: [RS033], [RS034].

d) Malfuncionamiento general del sistema

Debido a que nuestro sistema se embarcará en una aeronave debemos tener en cuenta como un funcionamiento erróneo de éste podría afectar a la resta de sistemas del avión. Anteriormente hemos visto como nuestro sistema estará en contacto con cableado del sistema de control y alimentación de los servomotores. Por este motivo deberemos considerar situaciones límite como puede ser un sobrecalentamiento del sistema. También es importante estudiar un posible desprendimiento de nuestro sistema dentro del cuerpo de la aeronave, lo que sin duda afectaría en gran medida a las actuaciones mecánicas de nuestro avión.

Al contemplar que el sobrecalentamiento o desprendimiento de nuestro sistema puede comprometer la integridad del avión, consideremos que el encapsulado de nuestro sistema tendrá un DAL igual al del servomotor. En nuestro caso se trata de un DAL A al no haber servomotores redundantes. Esto implica un DAL A en: [RS032], [RS033] y [RS034].

Reportaremos a diseño los requisitos derivados: [RS033], [RS034].

**[RS030] Evitar falsear la medida que el sistema de control recibe del potenciómetro.**

SW03-02

FUENTE: Realización del ZSA

[Diseño detallado] [NO IMPLEMENTADO]

**[RS031] El sistema implantado debe asegurar el correcto suministro de potencia al sistema de control.**

SW03-02

FUENTE: Realización del ZSA

[Diseño detallado] [NO IMPLEMENTADO]

**[RS032] La instalación de nuestro sistema no alterará a las condiciones de seguridad que ofrece el encapsulado del servo.**

SW03-02

FUENTE: Realización del ZSA

[Diseño detallado] [NO IMPLEMENTADO]

**[RS033] El encapsulado del sistema permitirá un sobrecalentamiento del sistema sin afectar a los sistemas colindantes.**

SW03-02

FUENTE: Realización del ZSA

[Diseño detallado] [NO IMPLEMENTADO]

**[RS034] La fijación del sistema a las distintas superficies debe soportar perturbaciones y gradientes de temperatura.**

SW03-02

FUENTE: Realización del ZSA

[Diseño detallado] [NO IMPLEMENTADO]

Tabla 1: Requisitos derivados de la elaboración del ZSA.



## 5 COMO EVITAREMOS LOS ANTERIORES PROBLEMAS POTENCIALES.

### a) Potenciómetro

Para evitar falsear la medida que el sistema de control recibe del potenciómetro colocaremos nuestro sistema en paralelo, asegurándonos mantener la caída de tensión entre los bornes. Además, se conectará a un amplificador operacional no inversor, para que la corriente de entrada a nuestro sistema sea igual a 0 y por tanto toda la corriente se quede en el servo.

### b) Encapsulado del servomotor.

Según la última revisión del documento de diseño, para asegurar que nuestro sistema no compromete en nada la integridad de la estructura interna del servomotor una vez pasado el cableado por la carcasa del servo sellaremos el agujero con una gota de silicona haciendo presión entre las dos partes que conforman la carcasa de nuestro servo. En la **Figura 14** podemos ver dos fotografías del resultado.

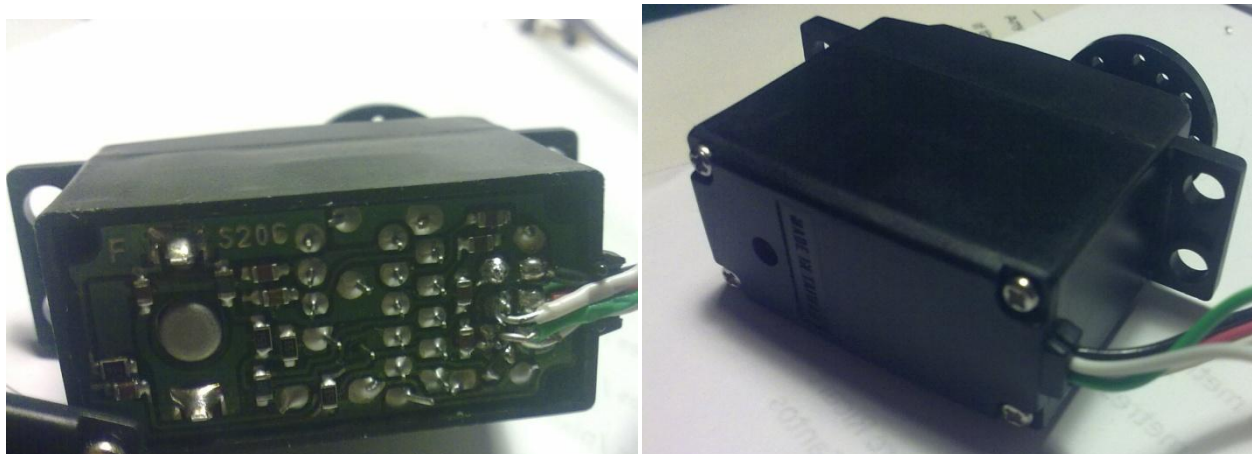


Figura 14: Cableado a extraer del servo.

### c) Encapsulado de nuestro sistema.

Aunque en diseño no se ha estudiado este apartado al habernos basado en un prototipo, desde safety para asegurar que nuestro sistema no supone ninguna traba a los sistemas mecánicos de la aeronave buscaremos un encapsulado compacto y adaptable a las diferentes exigencias de cada superficie de control. Se ha escogido una caja de PVC como la mostrada en la **Figura 15**.



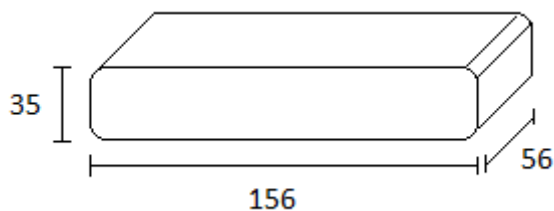


Figura 15: Dimensiones de la caja que contendrá el sistema

#### d) Malfuncionamiento general del sistema

La probabilidad de que se llegue a dar un funcionamiento anómalo tan catastrófico como para dañar los sistemas colindantes al nuestro es casi inexistente, aún así en el departamento de diseño se ha estimado positivo el uso de un encapsulado térmicamente aislante y autoextingible a más de aislante eléctrico tal y como podemos ver en la **Tabla 2**.

En cambio lo que si debemos estudiar con más detenimiento es el posible desprendimiento de nuestro sistema dentro del cuerpo del avión, ya que podría llegar a inutilizar las superficies de control. Por esto, y según la última revisión del documento de diseño, fijaremos nuestro sistema con silicona resistente a los cambios de temperatura. También se espera atenuar en cierta forma las vibraciones causadas por el funcionamiento normal de la aeronave.

PROPIEDADES PVC(POLICLORURO DE VINILO)
<ul style="list-style-type: none"> <li>• Rango de temperatura de trabajo -15°C +60°C.</li> <li>• Resistencia, rigidez y dureza mecánicas elevadas</li> <li>• Buen aislante eléctrico</li> <li>• Elevada resistencia a sustancias químicas</li> <li>• Autoextingible</li> <li>• Impermeable a gases y líquidos</li> <li>• Mínima absorción de agua</li> <li>• Resistente a la acción de hongos, bacterias, insectos y roedores</li> <li>• Fácil de pegar y soldar Resistente a la intemperie (sol, lluvia, viento y aire marino);</li> </ul>

Tabla 2: Propiedades del Policloruro de vinilo (PVC)



## 6 BIBLIOGRAFÍA

-SAE ARP4761, "Guidelines and methods for conducting the safety assessment process on civil airborne systems and equipment". (U.S.A., 1996)

NASA/TM-2007-214539, "Preliminary Considerations for Classifying Hazards of Unmanned Aircraft Systems" (U.S.A., 2007)

### DATASHEETS:

- "LM2397" URL: (<http://docs-europe.electrocomponents.com/webdocs/06e8/0900766b806e8404.pdf>)  
Fecha de consulta: 31/06/2009
- "WIZ810MJ" URL: ([http://www.circuitcellar.com/wiznet/WIZ810MJ%20Datasheet\\_V\\_1.0.pdf](http://www.circuitcellar.com/wiznet/WIZ810MJ%20Datasheet_V_1.0.pdf))  
Fecha de consulta: 31/06/2009
- "MSP430 2274 Family User's Guide" URL: (<http://focus.ti.com/lit/ug/slau144e/slau144e.pdf>)  
Fecha de consulta: 31/06/2009
- "MSP430 RF2500 Development Kit" URL: (<http://focus.ti.com/lit/ug/slau227e/slau227e.pdf>)  
Fecha de consulta: 31/06/2009



## 7 GLOSARIO

FHA: Functional Hazard Assessment

PSSA: Preliminary System Safety Assessment

ZSA: Zonal Safety Assessment

FTA: Fault Tree Analysis



# Diseño de Sistema Sensor Para superficies de UAV

---

Iss./Rev.: 12/07/2009

Date: 08/07/2009

[END OF DOCUMENT]

# ANEXO 10: Resumen de verificación

	Name	Signature	Date
Proyectista 1	Eloy Herrera		12/07/09
Proyectista 2	Víctor de los Santos		12/07/09
Director	Jorge Ramírez Alcantara		12/07/09

## Registro de cambios

Fecha	Cambio
10/07/09	-Se cambia la introducción por otra que ofrece una guía más detallada de cómo elaborar un PSSA
12/07/09	- Se cambia el formato. -Se eliminan requisitos de la verification matrix ya que no podían ser comprobados.

<b>1</b>	<b>INTRODUCCIÓN .....</b>	<b>6</b>
<b>2</b>	<b>MATRIZ DE VERIFICACIÓN .....</b>	<b>7</b>
<b>3</b>	<b>REPORTE DE PROBLEMAS .....</b>	<b>12</b>
<b>4</b>	<b>RECOPILACIÓN DE DOCUMENTOS CONCERNIENTES A LA VERIFICACIÓN .....</b>	<b>14</b>
4.1	Documentos fuente: .....	14
<b>5</b>	<b>BIBLIOGRAFÍA .....</b>	<b>15</b>
<b>6</b>	<b>GLOSARIO .....</b>	<b>16</b>

TABLA 1: MATRIZ DE VERIFICACIÓN .....	11
TABLA 2: EJEMPLO DE REPORTE DE PROBLEMA .....	12
TABLA 3: REPORTE DE PROBLEMA 01 .....	13
TABLA 4: REPORTE DE PROBLEMA 02 .....	13
TABLA 5: DOCUMENTOS UTILIZADOS COMO REFERENCIA .....	14
TABLA 6: DOCUMENTOS GENERADOS .....	14



FIGURA 1: PROCESO DE VERIFICACIÓN.....	6
--	---

## 1 INTRODUCCIÓN

Según el documento SAE ARP4754 el cuál hemos seguido como guía para la elaboración de este resumen de verificación, el propósito de la verificación es asegurar que todos los niveles de implementación cumplen con sus requisitos especificados. La verificación consiste en inspecciones, análisis, revisiones, tests i experiencia en servicio acorde a un plan de verificación.

Los procesos de verificación tratan de asegurar que todas las funciones han sido correctamente implementadas, que todos los requisitos han sido satisfechos y asegura que el análisis de safety sigue siendo válido para el sistema implementado.

El proceso de elaboración se compone de tres actividades distintas tal y como se puede observar en la **Figura 1**:

- Planning:** Planear los documentos que necesitaremos, la secuencia de actividades a realizar, los documentos que deberemos producir, la recopilación de la información necesaria , selección de actividades específicas y sus criterios de evaluación, y la generación de un hardware y software verificado.
- Actividades o pruebas de verificación:** Las pruebas de verificación son completamente dependientes una de ellas, tanto es así que no se procederá a realizar una prueba nueva hasta que la anterior cumpla con los parámetros de corrección establecidos.

Para realizar estas pruebas nos basaremos en tres métodos principales:

- Inspección y revisión:** Se consensuará si el sistema cumple con sus requisitos tales como su implementación física, si su diseño se corresponde a lo descrito en el documento de diseño y si cada uno de los elementos que componen el sistema cumplen a la vez con sus requisitos. Para éste tipo de tareas nos ayudaremos con unos checklist (Checklist) redactados por el equipo de verificación.
- Testeado:** Mediante el procedimiento de test evidenciaremos la repetida corrección en el tiempo de nuestro sistema. Demostrando así el cumplimiento de los requisitos. Usaremos un estricto criterio de Válido/Fallido. Para la realización de este tipo de pruebas deberemos consultar los documentos de test descritos por el equipo de verificación.
- Parecido/Experiencia:** Realmente la verificación del sistema total se basará en aportar evidencias de la corrección que nuestro sistema podrá tener al ser implementado en las diferentes superficies del Shadow o cualquier otra aeronave.

- Documentación:** Como documentos de verificación facilitaremos los checklist y documentos de test descritos anteriormente así como una matriz de verificación donde se incluirán los requisitos a verificar, la función asociada al requisito, en DAL (Development Assurance Level), los métodos aplicados en la verificación, la conclusión: válido/fallido y un resumen dónde anotar posibles eventualidades.

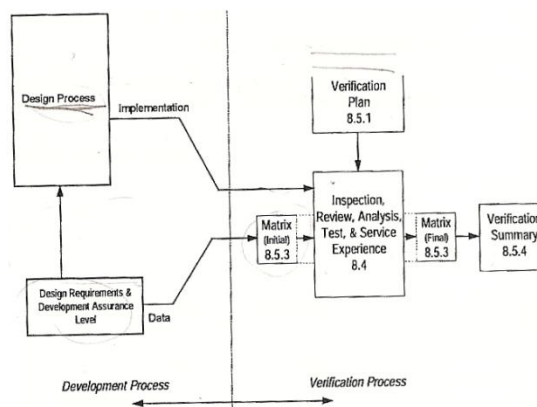


Figura 1: Proceso de verificación

## 2 MATRIZ DE VERIFICACIÓN

Una matriz de verificación (o verification matrix) u otro documento de guía equivalente deberá implementarse para seguir el proceso de verificación. El nivel de esta matriz suele depender del DAL del elemento o sistema a verificar. Estas matrices pueden tener el formato que el departamento de verificación elija, pero deben contener como mínimo:

- Los requisitos.
- Función asociada al requisito.
- DAL.
- Método aplicado para verificar el requisito.
- Conclusión (Pasado o fallido)
- Eventualidades ocurridas durante el proceso de verificación.

Los métodos aplicados para verificar un requisito podrán ser cualquiera de los mencionados anteriormente:

- a) **Inspección y revisión:** Consultar anexo (Checklist). Este anexo es una lista con diferentes aspectos sencillos a inspeccionar, usaremos un criterio de pasado o fallido en función si se ha podido observar que se cumple aquello enunciado por el checklist o no.
- b) **Testeado:** Consultar los anexos (XXX). Estos contienen unos procedimientos de test con unos pasos muy claros y sencillos a seguir y al final de los cuales deberemos decidir si el sistema pasa el test o no acorde a unas tolerancias dadas. Los tests tendrán la siguiente estructura:
  - Item testeado.
  - Equipo necesario para realizar el test.
  - Procedimiento del test
  - Tolerancias aceptadas.
- c) **Parecido/Experiencia:** Realmente la verificación del sistema total se basará en aportar evidencias de la corrección que nuestro sistema podrá tener al ser implementado en las diferentes superficies del Shadow o cualquier otra aeronave. No podemos establecer un ciclo acelerado en una cámara anicónica dónde probar que nuestro sistema funciona sin ningún incidente durante las horas que sería necesario para establecer que cumplimos un DAL A.

NOTA: En la **Tabla 1** podemos observar la tabla de verificación de nuestro sistema. En ella sólo hemos mencionado aquellos requisitos de más alto nivel y de los que podíamos crear unas pautas para su verificación.



Diseño de  
Sistema Sensor  
Para superficies  
de UAV

Iss./Rev.: 12/07/2009

Date: 10/07/2009

Requisito	Función asociada	DAL	Método de Verificación	Conclusión	Eventualidades
[R001]	Medir actuaciones timón de profundidad.	C	Test 01	Sólo testeado a nivel de prototipo. Resultado: Válido	
[R002]	Medir actuaciones timón de dirección.	C	Test 01	Sólo testeado a nivel de prototipo. Resultado: Válido	
[R003]	Medir actuaciones timón del alerón izquierdo.	C	Test 01	Sólo testeado a nivel de prototipo. Resultado: Válido	
[R004]	Medir actuaciones timón del alerón derecho.	C	Test 01	Sólo testeado a nivel de prototipo. Resultado: Válido	
[R006]	El sistema no debe sobrepasar el peso máximo permitido del avión	C	Checklist	Válido	
[R007]	El sistema debe caber en las zonas donde se encuentran los sistemas de los que se quiere obtener datos.	C	Checklist	No comprobado	El prototipo no tiene las medidas definitivas



## Diseño de Sistema Sensor Para superficies de UAV

Iss./Rev.: 12/07/2009

Date: 10/07/2009

[R008]	Independencia entre los distintos subsistemas.	C	Parecido/Experiencia	No comprobado	No se ha podido probar con los sistemas deinitivos.
[R009]	Medir actuaciones timón de profundidad.	C	Test 01	Sólo testeado a nivel de prototipo. Resultado: Válido	
[R010]	Medir actuaciones timón de dirección.	C	Test 01	Sólo testeado a nivel de prototipo. Resultado: Válido	
[R011]	Medir actuaciones timón del alerón izquierdo.	C	Test 01	Sólo testeado a nivel de prototipo. Resultado: Válido	
[R012]	Medir actuaciones timón del alerón derecho.	C	Test 01	Sólo testeado a nivel de prototipo. Resultado: Válido	
[R014]	El sistema transmitirá los datos a través de ETHERNET	C	Checklist	Válido	Consultar reporte de problemas. (Problema 01)
[R017]	El sistema debe poder transmitir a través de una conexión SERIE	C	Checklist	Válido	Consultar reporte de problemas. (Problema 02)



Diseño de  
Sistema Sensor  
Para superficies  
de UAV

Iss./Rev.: 12/07/2009

Date: 10/07/2009

[RS030]	Evitar falsear la medida que el sistema de control recibe del potenciómetro.	A	Test 02	Válido
[RS031]	El sistema implantado debe asegurar el correcto suministro de potencia al sistema de control.	A	Checklist	Válido
[RS032]	La instalación de nuestro sistema no alterará a las condiciones de seguridad que ofrece el encapsulado del servo.	A	Checklist	Válido
[RS033]	El encapsulado del sistema permitirá un sobrecalentamiento del sistema sin afectar a los sistemas colindantes..	A	Test 03	No comprobado
[RS034]	La fijación del sistema a las distintas superficies debe soportar perturbaciones y gradientes de temperatura.	A	Checklist	No comprobado
[RS035]	La señal no se verá afectada por el ruido generado en el avión.	C	Test 04	No comprobado



Diseño de  
Sistema Sensor  
Para superficies  
de UAV

Iss./Rev.: 12/07/2009  
Date: 10/07/2009

[RS036]	El sistema no afectará al correcto comportamiento mecánico de las superficies de control	A	Checklist	No comprobado
[RS037]	El sistema no interferirá en la aerodinámica de la aeronave	A	Checklist	No comprobado

Tabla 1: Matriz de verificación.

### 3 REPORTE DE PROBLEMAS

Otra de las partes que debe contener un informe de verificación según consta en el ARP4761 es un reporte con todos los problemas que nos hayamos encontrado durante el proceso de verificación. Cuando un test tenga un resultado negativo (Fallido), el error que lo haya ocasionado será reportado al departamento de diseño dónde deberán encontrarle una solución.

Los errores se reportarán con un formato como el que podemos observar en la Tabla 2.

Requisito:	Fecha problema:	
	Fecha solución:	
Descripción		
Implicación de seguridad:		

Tabla 2: Ejemplo de reporte de problema

Los parámetros a incluir en este reporte son:

- **Requisito:** Identificador y explicación del requisito de diseño o safety que no se ha conseguido superar
- **Fecha problema:** Día en el que nos hemos encontrado este fallo.
- **Fecha solución:** Día en el que el departamento de diseño nos comunica mediante informe que el problema ha sido solucionado.  
 IMPORTANTE: NO es el día que el departamento de verificación constatar que efectivamente el error ha sido superado.  
 Un recuadro donde el departamento de diseño incluirá una “V” de válido cuando haya podido constatar que efectivamente el error ha sido superado.
- **Descripción:** Describiremos el problema hallado indicando el momento en el que fuimos conscientes de éste y en qué consiste. Aportaremos a más los valores obtenidos y los que se esperaban obtener para facilitar en medida de lo posible la faena del equipo de diseño.
- **Implicación de seguridad:** Las implicaciones que este error tendría en el global del sistema y en la aeronave en caso de no haber sido corregirse.





### Reporte de problemas:

Requisito:	Fecha problema:1/07/2009	
[R014]	Fecha solución:3/07/2009	
Descripción:		
No se reciben los datos mediante el puerto Ethernet.		
Se solucionó revisando los datasheets del microcontrolador y el módulo ethernet y cambiando el software porque había un problema en la configuración.		
Implicación de seguridad:		
Ninguna, como se explica en el FHA la monitorización del sistema no es una función crítica y por lo tanto cadece de importancia en un estudio de safety.		

Tabla 3: Reporte de problema 01

Requisito:	Fecha problema:1/07/2009	
[R017]	Fecha solución:1/07/2009	
Descripción		
No se reciben los datos mediante el puerto serie.		
Se solucionó revisando los datasheets del microcontrolador y cambiando el software porque había un problema en la configuración.		
Implicación de seguridad:		
Ninguna, como se explica en el FHA la monitorización del sistema no es una función crítica y por lo tanto cadece de importancia en un estudio de safety.		

Tabla 4: Reporte de problema 02

## 4 RECOPIACIÓN DE DOCUMENTOS CONCERNIENTES A LA VERIFICACIÓN

Por último, para completar el verification summary se nos pide que hagamos un recopilatorio con toda aquella información fuente que hemos utilizado para la redacción de éste y de todos aquellos documentos que hayamos producido.

En la preparación de las pruebas de verificación, que empiezan ya a definirse al empezar el diseño arquitectural, se utilizan gran cantidad de documentos que anteriormente hayamos generado. En nuestro caso estos documentos quedan recogidos en la **Tabla 5**.

Es cierto también que en el proceso de verificación se produce gran cantidad de documentación, la mayoría de ella relacionada con las pruebas a las que debemos someter tanto a hardware como software para verificar su validez y su corrección respecto a los requisitos establecidos. Los documentos que nosotros hemos generado en la elaboración de este reporte o resumen están citados en la **Tabla 6**.

### 4.1 Documentos fuente:

Documentos utilizados como referencia
Documento de diseño arquitectural
Documento de diseño
FHA
PSSA
ZSA
Listado de requisitos

Tabla 5: Documentos utilizados como referencia

Documentos generados
Checklist
Test 01
Test 02
Test 03
Test 04

Tabla 6: Documentos generados



## 5 BIBLIOGRAFÍA

- SAE ARP4761, "Guidelines and methods for conducting the safety assessment process on civil airborne systems and equipment". U.S.A., 1996



## 6 GLOSARIO

FHA: Functional Hazard Assessment

PSSA: Preliminary System Safety Assessment

ZSA: Zonal Safety Assessment

FTA: Fault Tree Analysis

SSA: System Safety Analysis



Diseño de  
Sistema Sensor  
Para superficies  
de UAV

Iss./Rev.: 12/07/2009

Date: 10/07/2009

---

[END OF DOCUMENT]

<b>Test 01</b>
02/07/2009

---

**ITEMS TESTEADOS**

- Prototipo sistema sensor.

---

**EQUIPO**

- Prototipo sistema sensor
- Superficie prototipo.
- Estación receptora

---

**PROCEDIMIENTO**

- Mediremos los ángulos máximos y mínimos de recorrido del prototipo.
- Una vez todo el software y hardware esté operativo efectuaremos diversos cambios en la posición de la superficie prototipo y verificaremos que los datos recibidos son correctos y se corresponden al movimiento efectuado.
- Para asegurar la corrección de los datos obtenidos mediremos el ángulo al que sometemos a la superficie y lo compararemos con el que nos da nuestro sistema.

---

**TOLERÁNCIAS ACEPTADAS**

La diferencia máxima aceptada por el equipo de verificación es de 12º.

<b>Test 01</b>
02/07/2009

---

**RESULTADOS**

El sistema mide el ángulo con un error menor a 12<sup>a</sup>. En los resultados no se ha obtenido una deriva mayor de 6<sup>o</sup>.

**Test 02**

02/07/2009

**ITEMS TESTEADOS**

---

- Potenciómetro

**EQUIPO**

---

- Potenciómetro del servomotor.
- Voltímetro.
- Prototipo

**PROCEDIMIENTO**

---

- Se medirán las tensiones del potenciómetro del servo con un voltímetro para diferentes posiciones antes de conectar el potenciómetro al sistema.
- Se realizarán las mismas medidas para las diferentes posiciones pero esta vez con el sistema conectado y en funcionamiento.
- Se compararán los resultados obtenidos.

**TOLERÁNCIAS ACEPTADAS**

---

El margen de error será la resolución del voltímetro y de la posición del servo.



<b>Test 02</b>
02/07/2009

---

**RESULTADOS**

Los valores comparados coinciden (la señal no queda falseada).

<b>Test 03</b>
02/07/2009

---

**ITEMS TESTEADOS**

- Sistema

---

**EQUIPO**

- Sistema sensor
- Sensor de temperatura

---

**PROCEDIMIENTO**

- Se medirá la temperatura externa del sistema durante su funcionamiento

---

**TOLERÁNCIAS ACEPTADAS**

Máximo 60 °C

<b>Test 03</b>
02/07/2009

---

**RESULTADOS**

El exterior del sistema no supera los 60 ºC

<b>Test 04</b>
02/07/2009

---

**ITEMS TESTEADOS**

- Sistema

---

**EQUIPO**

- Sistema
- UAV de destino

---

**PROCEDIMIENTO**

- Todos los sistemas del UAV se pondrán en funcionamiento tal y como estarían en vuelo.
- Se pondrá el sistema en funcionamiento y se comprobará que los valores que salen del sistema coinciden con la posición real.

---

**TOLERÁNCIAS ACEPTADAS**

El error máximo será la resolución del sistema.

<b>Checklist</b>	<b>Checklist</b>
02/07/2009	

Requisitos	Conclusión
El sistema muestra las actuaciones realizadas por la superficie prototipo	✓
El sistema no pesa más del payload máximo.	✓
El sistema transmite por Ethernet	✓
El sistema transimite por série	✓
El microcontrolador viene de serie con un amplificador operacional.	✓
La instalación de nuestro sistema no altera el correcto cierre del encapsulado del servo.	✓
El sistema no afecta a las actuaciones mecánicas del servo.	
El sistema se instalará en los huecos asignados y no modificará ninguna superficie que este en contacto con el flujo de aire.	

[END OF DOCUMENT]

## ANEXO 11: Lista de requisitos

	Name	Signature	Date
Proyectista 1	Eloy Herrera		07/07/09
Proyectista 2	Víctor de los Santos		07/07/09
Director	Jorge Ramírez Alcantara		07/07/09



Diseño de  
Sistema Sensor  
Para superficies  
de UAV

Iss./Rev.: D01  
Date: 07/07/2009

---

## Contenido

<b>1</b>	<b>REQUISITOS DE DISEÑO .....</b>	<b>3</b>
<b>2</b>	<b>REQUISITOS DE SAFETY .....</b>	<b>9</b>



## 1 REQUISITOS DE DISEÑO

REQUISITO	TRAZABLE A
<p>[R001] El sistema mostrará las actuaciones del timón de profundidad El sistema será capaz de medir, tratar, enviar y mostrar al usuario las actuaciones del timón de profundidad, entendiendo por actuaciones el desplazamiento angular realizado por dicha superficie. FUENTE: Reunión con el cliente [Diseño arquitectural] [NO IMPLEMENTADO]</p>	
<p>[R002] El sistema mostrará las actuaciones del timón de dirección El sistema será capaz de medir, tratar, enviar y mostrar al usuario las actuaciones del timón de cola, entendiendo por actuaciones el desplazamiento angular realizado por dicha superficie. FUENTE: Reunión con el cliente [Diseño arquitectural] [NO IMPLEMENTADO]</p>	
<p>[R003] El sistema mostrará las actuaciones del alerón izquierdo El sistema será capaz de medir, tratar, enviar y mostrar al usuario las actuaciones del alerón izquierdo, entendiendo por actuaciones el desplazamiento angular realizado por dicha superficie. FUENTE: Reunión con el cliente [Diseño arquitectural] [NO CUMPLIDO]</p>	
<p>[R004] El sistema mostrará las actuaciones del alerón derecho El sistema será capaz de medir, tratar, enviar y mostrar al usuario las actuaciones del alerón derecho, entendiendo por actuaciones el desplazamiento angular realizado por dicha superficie. FUENTE: Reunión con el cliente [Diseño arquitectural] [NO CUMPLIDO]</p>	
<p>[R005] El sistema mostrará las velocidad de giro del motor El sistema será capaz de medir, tratar, enviar y mostrar al usuario la velocidad angular del motor. FUENTE: Reunión con el cliente [Diseño arquitectural] [DESCARTADO]</p>	
<p>[R006] El sistema no debe sobrepasar el peso máximo permitido del avión El sistema será ligero para no comprometer las performances del avión. FUENTE: Estudio del sistema [Diseño arquitectural] [IMPLEMENTADO]</p>	<p>[R001] [R002] [R003] [R004]</p>
<p>[R007] El sistema debe caber en las zonas donde se encuentran los sistemas de los que se quiere obtener datos. El sistema será pequeño debido al reducido espacio del que dispone el avión para el sistema. FUENTE: Estudio del sistema [Diseño arquitectural] [NO CUMPLIDO]</p>	<p>[R001] [R002] [R003] [R004]</p>

<p>[R008] Los diferentes subsistemas de medición (S1, S2, S3 y S4) deben trabajar de forma independiente Los subsistemas de medición mencionados funcionarán de forma independiente entre ellos y por tanto, en caso de fallo de uno, los otros seguirán funcionando. FUENTE: Estudio del sistema [Diseño arquitectural] [IMPLEMENTADO]</p>	<p>[R001] [R002] [R003] [R004]</p>
<p>[R009] El subsistema S1 medirá y transmitirá la posición angular del timón de profundidad El subsistema medirá la posición (en grados) del timón de profundidad y lo transmitirá. FUENTE: Diseño arquitectural del sistema [Diseño arquitectural] [NO IMPLEMENTADO]</p>	<p>[R001]</p>
<p>[R010] El subsistema S2 medirá y transmitirá la posición angular del timón de dirección El subsistema medirá la posición (en grados) del timón de profundidad y lo transmitirá. FUENTE: Diseño arquitectural del sistema [Diseño arquitectural] [NO IMPLEMENTADO]</p>	<p>[R002]</p>
<p>[R011] El subsistema S3 medirá y transmitirá la posición angular del alerón izquierdo El subsistema medirá la posición (en grados) del timón de profundidad y lo transmitirá. FUENTE: Diseño arquitectural del sistema [Diseño arquitectural] [NO CUMPLIDO]</p>	<p>[R003]</p>
<p>[R012] El subsistema S4 medirá y transmitirá la posición angular del alerón derecho El subsistema medirá la posición (en grados) del timón de profundidad y lo transmitirá. FUENTE: Diseño arquitectural del sistema [Diseño arquitectural] [NO CUMPLIDO]</p>	<p>[R004]</p>
<p>[R013] El sistema debe utilizar como sensor el potenciómetro interno del servo El sistema utilizará el sensor interno del servo, un potenciómetro, para la medida de los datos FUENTE: Diseño detallado del sistema [Diseño detallado] [IMPLEMENTADO]</p>	<p>[R009] [R010] [R011] [R012]</p>
<p>[R014] El sistema transmitirá los datos a través de ETHERNET El sistema transmitirá los datos a través de ETHERNET para poder así funcionar con la futura red MAREA del UAV FUENTE: Diseño arquitectural del sistema [Diseño arquitectural] [IMPLEMENTADO]</p>	<p>[R009] [R010] [R011] [R012]</p>

<p>[R015] El rango dinámico del sistema será igual o mayor a 27.6 dB Tras el estudio del sistema, se obtiene que el rango dinámico debe ser mayor a 27.6 dB, que es el rango dinámico con la peor resolución del UAV FUENTE: Diseño arquitectural del sistema [Diseño arquitectural] [IMPLEMENTADO]</p>	<p>[R009] [R010] [R011] [R012]</p>
<p>[R017] El sistema debe poder transmitir a través de una conexión SERIE El sistema podrá transmitir los datos a través de una conexión SERIE que se utilizará principalmente como conexión de pruebas FUENTE: Diseño arquitectural del sistema [Diseño detallado] [IMPLEMENTADO]</p>	<p>[R009] [R010] [R011] [R012]</p>
<p>[R018] El sistema debe tener una entrada constante de 3.3V Las distintas partes del sistema funcionan con una tensión de 3.3V, y por tanto debe asegurarse esta entrada FUENTE: Diseño detallado del sistema [Diseño detallado] [IMPLEMENTADO]</p>	<p>[R009] [R010] [R011] [R012]</p>
<p>[R019] El sistema debe ir ensamblado en una única placa de circuito impreso El sistema irá todo formado por una única placa de circuito impreso para darle mayor robustez al sistema FUENTE: Diseño detallado del sistema [Diseño detallado] [IMPLEMENTADO]</p>	<p>[R009] [R010] [R011] [R012]</p>
<p>[R020] El sistema de alimentación deberá tener un regulador LDO El sistema utilizará un regulador LDO para asegurar los 3.3 V de tensión necesarios para el funcionamiento del sistema FUENTE: Diseño detallado del sistema [Diseño detallado] [IMPLEMENTADO]</p>	<p>[R018]</p>
<p>[R021] El LDO debe permitir un paso de corriente de mínimo 200 mA Los componentes de cada subsistema consumen una corriente que puede alcanzar los 200 mA. Para ello el LDO debe permitir el paso de un mínimo de 200 mA. FUENTE: Diseño detallado del sistema [Diseño detallado] [IMPLEMENTADO]</p>	<p>[R021]</p>
<p>[R022] El microcontrolador del subsistema debe tener un convertor analógico-digital de mínimo 5 bits integrado Para evitar complicar el sistema, se escogerá un microcontrolador que incorpore un convertor analógico-digital FUENTE: Diseño detallado del sistema [Diseño detallado] [IMPLEMENTADO]</p>	<p>[R009] [R010] [R011] [R012]</p>

<p>[R023] El sensor del sistema debe estar conectado a un amplificador operacional El sensor estará conectado a un amplificador operacional para evitar falsear la señal que emite el sensor al servo FUENTE: Diseño detallado del sistema [Diseño detallado] [IMPLEMENTADO]</p>	<p>[R009] [R010] [R011] [R012]</p>
<p>[R024] El microcontrolador del subsistema debe tener un amplificador operacional integrado El microcontrolador utilizado en el sistema debe incorporar un amplificador operacional FUENTE: Diseño detallado del sistema [Diseño detallado] [IMPLEMENTADO]</p>	<p>[R023]</p>
<p>[R025] El microcontrolador debe tener dos USCI El microcontrolador debe tener dos Interfaces de comunicación serie universal, para poder realizar la conexión serie y la conexión ETHERNET FUENTE: Diseño detallado del sistema [Diseño detallado] [IMPLEMENTADO]</p>	<p>[R009] [R010] [R011] [R012]</p>
<p>[R026] El módulo ETHERNET debe poder conectarse al microcontrolador por un USCI El módulo ETHERNET debe poder conectarse al microcontrolador a través de una de los dos interfaces de comunicación serie universal de éste para su control y correcto funcionamiento FUENTE: Diseño detallado del sistema [Diseño detallado] [IMPLEMENTADO]</p>	<p>[R025]</p>
<p>[R027] Todo el software del sistema debe estar concentrado en el microcontrolador El microcontrolador debe ser la única parte del sistema programable y que necesite ser controlada mediante software FUENTE: Diseño detallado del sistema [Diseño detallado] [IMPLEMENTADO]</p>	<p>[R009] [R010] [R011] [R012]</p>
<p>[R028] El software debe poder adaptarse fácilmente a cualquier superficie que se quiera monitorizar La adaptación del software debe ser sencilla y solo necesitar pequeños ajustes para la adaptación a cualquier superficie que se quiera monitorizar FUENTE: Diseño detallado del sistema [Diseño detallado] [IMPLEMENTADO]</p>	<p>[R027]</p>
<p>[R029] El software debe estar dividido en funciones En función de para que sirve cada parte del software, se debe dividir en funciones FUENTE: Diseño detallado del sistema [Diseño detallado] [IMPLEMENTADO]</p>	<p>[R027]</p>

<p>[R030] La función 1 del software debe controlar la recepción de los datos del sensor La función 1 controlará mediante software los accesos y las conexiones entre el microcontrolador y el sensor FUENTE: Diseño detallado del sistema [Diseño detallado] [IMPLEMENTADO]</p>	[R029]
<p>[R031] La función 2 del software debe controlar la amplificación de la señal La función 2 controlará el amplificador operacional integrado en el software FUENTE: Diseño detallado del sistema [Diseño detallado] [IMPLEMENTADO]</p>	[R029]
<p>[R032] La función 3 del software debe controlar la conversión Analógico-digital La función 3 controlará la conversión de la señal de analógico a digital FUENTE: Diseño detallado del sistema [Diseño detallado] [IMPLEMENTADO]</p>	[R029]
<p>[R033] La función 4 debe procesar los datos para obtener la información deseada La función 4 será la encargada de aplicar las ecuaciones necesarias para obtener la información de la posición de la superficie deseada FUENTE: Diseño detallado del sistema [Diseño detallado] [IMPLEMENTADO]</p>	[R029]
<p>[R034] La función 5 debe controlar la conexión SERIE La conexión SERIE del sistema debe estar controlada mediante la función 5 FUENTE: Diseño detallado del sistema [Diseño detallado] [IMPLEMENTADO]</p>	[R029]
<p>[R035] La función 6 debe controlar la conexión ETHERNET La conexión ETHERNET se controlará a través de la función 6 FUENTE: Diseño detallado del sistema [Diseño detallado] [IMPLEMENTADO]</p>	[R029]
<p>[R036] La ganancia del amplificador debe ser <math>G=1</math> La ganancia del amplificador debe ser <math>G=1</math>, para que la señal de entrada y salida de éste sea la misma FUENTE: Diseño detallado del sistema [Diseño detallado] [IMPLEMENTADO]</p>	[R031]
<p>[R037] El amplificador debe estar configurado como NO-INVERSOR El amplificador necesitado en el sistema es NO-INVERSOR y por tanto debe estar configurado como tal FUENTE: Diseño detallado del sistema [Diseño detallado] [IMPLEMENTADO]</p>	[R031]



## Diseño de Sistema Sensor Para superficies de UAV

Iss./Rev.: D01  
Date: 07/07/2009

[R038] Las variables que definen cada superficie deben ir en un archivo de cabecera Para poder modificar fácilmente cada variable que define a las superficies, se utilizará un archivo de cabecera en vez de grabarlas directamente en el archivo de código FUENTE: Diseño detallado del sistema [Diseño detallado] [IMPLEMENTADO]	[R028]
[R039] El sistema debe transmitir un mensaje formado por una cabecera, un identificador y la posición de la superficie El mensaje transmitido debe contener una cabecera por definir, un identificador de la superficie monitorizada y la posición de la superficie FUENTE: Diseño detallado del sistema [Diseño detallado] [IMPLEMENTADO]	[R033] [R034] [R035]
[R040] El sistema debe transmitir un indicativo de error si ha detectado un error en el procesamiento de datos El sistema emitirá un mensaje con un indicativo de error si ha detectado algún fallo en el procesamiento de datos FUENTE: Diseño detallado del sistema [Diseño detallado] [IMPLEMENTADO]	[R033] [R034] [R035]
[R041] El sistema debe transmitir a través de ETHERNET usando un protocolo UDP El protocolo utilizado para la transmisión por ETHERNET será UDP FUENTE: Reunión con el coordinador de la red MAREA del UAS [Diseño detallado] [IMPLEMENTADO]	[R035]
[R042] El sistema debe realizar un broadcast de la información a la red ETHERNET Se realizará un broadcast por toda la red permitiendo coger la información al usuario que lo desee FUENTE: Reunión con el coordinador de la red MAREA del UAS [Diseño detallado] [IMPLEMENTADO]	[R041]

## 2 REQUISITOS DE SAFETY

[RS000] La probabilidad de pérdida del sistema debe estar clasificada como "Major Failure Condition" FUENTE: Realización del FHA [Diseño arquitectural] [IMPLEMENTADO]	[RS000]
[RS001] La probabilidad del fallo 1 debe estar clasificada como "Major Failure Condition" FUENTE: Realización del FHA [Diseño arquitectural] [IMPLEMENTADO]	[RS000]
[RS002] La probabilidad del fallo 2 debe estar clasificada como "Major Failure Condition" FUENTE: Realización del FHA [Diseño arquitectural] [IMPLEMENTADO]	[RS000]
[RS003] La probabilidad del fallo 3 debe estar clasificada como "Major Failure Condition" FUENTE: Realización del FHA [Diseño arquitectural] [IMPLEMENTADO]	[RS000]
[RS004] La probabilidad del fallo 4 debe estar clasificada como "Major Failure Condition" FUENTE: Realización del FHA [Diseño arquitectural] [IMPLEMENTADO]	[RS000]
[RS005] La probabilidad del fallo 5 debe estar clasificada como "Major Failure Condition" FUENTE: Realización del FHA [Diseño arquitectural] [IMPLEMENTADO]	[RS000]
[RS006] La probabilidad de fallo en el valor resistivo del potenciómetro debe estar clasificada como "Major Failure Condition" FUENTE: Realización del PSSA [Diseño detallado] [IMPLEMENTADO]	[RS000] [RS001]
[RS007] La probabilidad de fallo en el procesamiento de señal realizado por el microprocesador debe estar clasificada como "Major Failure Condition" FUENTE: Realización del PSSA. [Diseño detallado] [IMPLEMENTADO]	[RS000] [RS001]
[RS008] La probabilidad de fallo en la transmisión de la señal hasta la emisora del "Shadow" (Cableado) debe estar clasificada como "Major Failure Condition" FUENTE: Realización del PSSA. [Diseño detallado] [IMPLEMENTADO]	[RS000] [RS001]

<p>[RS009] La probabilidad de fallo en la recepción de la señal transmitida por la emisora del "Shadow" debe estar clasificada como "Major Failure Condition"</p> <p>FUENTE: Realización del PSSA.</p> <p>[Diseño detallado] [IMPLEMENTADO]</p>	<p>[RS000] [RS001]</p>
<p>[RS010] La probabilidad de fallo en el procesado de la señal por el hardware del equipo de tierra debe estar clasificada como "Major Failure Condition"</p> <p>FUENTE: Realización del PSSA.</p> <p>[Diseño detallado] [IMPLEMENTADO]</p>	<p>[RS000] [RS001]</p>
<p>[RS011] La probabilidad de fallo en el procesado de la señal por el software del equipo de tierra debe estar clasificada como "Major Failure Condition"</p> <p>FUENTE: Realización del PSSA.</p> <p>[Diseño detallado] [IMPLEMENTADO]</p>	<p>[RS000] [RS001]</p>
<p>[RS012] La probabilidad de fallo en el valor resistivo del potenciómetro debe estar clasificada como "Major Failure Condition"</p> <p>FUENTE: Realización del PSSA</p> <p>[Diseño detallado] [IMPLEMENTADO]</p>	<p>[RS000] [RS002]</p>
<p>[RS013] La probabilidad de fallo en el procesado de señal realizado por el microprocesador debe estar clasificada como "Major Failure Condition"</p> <p>FUENTE: Realización del PSSA.</p> <p>[Diseño detallado] [IMPLEMENTADO]</p>	<p>[RS000] [RS002]</p>
<p>[RS014] La probabilidad de fallo en la transmisión de la señal hasta la emisora del "Shadow" (Cableado) debe estar clasificada como "Major Failure Condition"</p> <p>FUENTE: Realización del PSSA.</p> <p>[Diseño detallado] [IMPLEMENTADO]</p>	<p>[RS000] [RS002]</p>
<p>[RS015] La probabilidad de fallo en la recepción de la señal transmitida por la emisora del "Shadow" debe estar clasificada como "Major Failure Condition"</p> <p>FUENTE: Realización del PSSA.</p> <p>[Diseño detallado] [IMPLEMENTADO]</p>	<p>[RS000] [RS002]</p>
<p>[RS016] La probabilidad de fallo en el procesado de la señal por el hardware del equipo de tierra debe estar clasificada como "Major Failure Condition"</p> <p>FUENTE: Realización del PSSA.</p> <p>[Diseño detallado] [IMPLEMENTADO]</p>	<p>[RS000] [RS002]</p>





## Diseño de Sistema Sensor Para superficies de UAV

Iss./Rev.: D01  
Date: 07/07/2009

[RS017] La probabilidad de fallo en el procesamiento de la señal por el software del equipo de tierra debe estar clasificada como "Major Failure Condition" FUENTE: Realización del PSSA. [Diseño detallado] [IMPLEMENTADO]	[RS000] [RS002]
[RS018] La probabilidad de fallo en el valor resistivo del potenciómetro debe estar clasificada como "Major Failure Condition" FUENTE: Realización del PSSA [Diseño detallado] [IMPLEMENTADO]	[RS000] [RS003]
[RS019] La probabilidad de fallo en el procesamiento de señal realizado por el microprocesador debe estar clasificada como "Major Failure Condition" FUENTE: Realización del PSSA. [Diseño detallado] [IMPLEMENTADO]	[RS000] [RS003]
[RS020] La probabilidad de fallo en la transmisión de la señal hasta la emisora del "Shadow" (Cableado) debe estar clasificada como "Major Failure Condition" FUENTE: Realización del PSSA. [Diseño detallado] [IMPLEMENTADO]	[RS000] [RS003]
[RS021] La probabilidad de fallo en la recepción de la señal transmitida por la emisora del "Shadow" debe estar clasificada como "Major Failure Condition" FUENTE: Realización del PSSA. [Diseño detallado] [IMPLEMENTADO]	[RS000] [RS003]
[RS022] La probabilidad de fallo en el procesamiento de la señal por el hardware del equipo de tierra debe estar clasificada como "Major Failure Condition" FUENTE: Realización del PSSA. [Diseño detallado] [IMPLEMENTADO]	[RS000] [RS003]
[RS023] La probabilidad de fallo en el procesamiento de la señal por el software del equipo de tierra debe estar clasificada como "Major Failure Condition" FUENTE: Realización del PSSA. [Diseño detallado] [IMPLEMENTADO]	[RS000] [RS003]
[RS024] La probabilidad de fallo en el valor resistivo del potenciómetro debe estar clasificada como "Major Failure Condition" FUENTE: Realización del PSSA [Diseño detallado] [IMPLEMENTADO]	[RS000] [RS004]
[RS025] La probabilidad de fallo en el procesamiento de señal realizado por el microprocesador debe estar clasificada como "Major Failure Condition" FUENTE: Realización del PSSA. [Diseño detallado] [IMPLEMENTADO]	[RS000] [RS004]

<p>[RS026] La probabilidad de fallo en la transmisión de la señal hasta la emisora del "Shadow" (Cableado) debe estar clasificada como "Major Failure Condition"</p> <p>FUENTE: Realización del PSSA.</p> <p>[Diseño detallado] [IMPLEMENTADO]</p>	<p>[RS000] [RS004]</p>
<p>[RS027] La probabilidad de fallo en la recepción de la señal transmitida por la emisora del "Shadow" debe estar clasificada como "Major Failure Condition"</p> <p>FUENTE: Realización del PSSA.</p> <p>[Diseño detallado] [IMPLEMENTADO]</p>	<p>[RS000] [RS004]</p>
<p>[RS028] La probabilidad de fallo en el procesado de la señal por el hardware del equipo de tierra debe estar clasificada como "Major Failure Condition"</p> <p>FUENTE: Realización del PSSA.</p> <p>[Diseño detallado] [IMPLEMENTADO]</p>	<p>[RS000] [RS004]</p>
<p>[RS029] La probabilidad de fallo en el procesado de la señal por el software del equipo de tierra debe estar clasificada como "Major Failure Condition"</p> <p>FUENTE: Realización del PSSA.</p> <p>[Diseño detallado] [IMPLEMENTADO]</p>	<p>[RS000] [RS004]</p>
<p>[RS030] Evitar falsear la medida que el sistema de control recibe del potenciómetro.</p> <p>FUENTE: Realización del ZSA</p> <p>[Diseño detallado] [IMPLEMENTADO]</p>	
<p>[RS031] El sistema implantado debe asegurar el correcto suministro de potencia al sistema de control.</p> <p>FUENTE: Realización del ZSA</p> <p>[Diseño detallado] [IMPLEMENTADO]</p>	
<p>[RS032] La instalación de nuestro sistema no alterará a las condiciones de seguridad que ofrece el encapsulado del servo.</p> <p>FUENTE: Realización del ZSA</p> <p>[Diseño detallado] [IMPLEMENTADO]</p>	
<p>[RS033] El encapsulado del sistema permitirá un sobrecalentamiento del sistema sin afectar a los sistemas colindantes.</p> <p>FUENTE: Realización del ZSA</p> <p>[Diseño detallado] [NO IMPLEMENTADO]</p>	
<p>[RS034] La fijación del sistema a las distintas superficies debe soportar perturbaciones y gradientes de temperatura.</p> <p>FUENTE: Realización del ZSA</p> <p>[Diseño detallado] [NO IMPLEMENTADO]</p>	<p>[RS033]</p>



## Diseño de Sistema Sensor Para superficies de UAV

Iss./Rev.: D01

Date: 07/07/2009

[RS035] La señal no se verá afectada por el ruido generado en el avión. FUENTE: Realización del PSSA [Diseño detallado] [IMPLEMENTADO]	
[RS036] El sistema no afectará al correcto comportamiento mecánico de las superficies de control FUENTE: Realización del PSSA. [Diseño detallado] [NO IMPLEMENTADO]	
[RS037] El sistema no interferirá en la aerodinámica de la aeronave FUENTE: Realización del PSSA. [Diseño detallado] [NO IMPLEMENTADO]	



Diseño de  
Sistema Sensor  
Para superficies  
de UAV

---

Iss./Rev.: D01

Date: 07/07/2009

[END OF DOCUMENT]

## ANEXO 12: Código del Software

	Name	Signature	Date
Proyectista 1	Eloy Herrera		07/07/09
Proyectista 2	Víctor de los Santos		07/07/09
Director	Jorge Ramírez Alcantara		07/07/09

## CONTENIDO

<b>1</b>	<b>CÓDIGO.C .....</b>	<b>3</b>
<b>2</b>	<b>SISTEMA.H.....</b>	<b>10</b>
<b>3</b>	<b>ETHERNET.H .....</b>	<b>11</b>
<b>4</b>	<b>W5100REG.H .....</b>	<b>13</b>
<b>5</b>	<b>CALIBRACION.C .....</b>	<b>17</b>

## 1 CÓDIGO.C

```
#include "msp430x22x4.h"
#include "sistema.h"
#include "w5100reg.h"
#include "ethernet.h"

int Gs0_rx_base = 0x6000;

void w5100_init(void);
void w5100_write(int adres1,int adres2,int value);
void w5100_udp(char* string);
int w5100_read(int adres1,int adres2);
void w5100_data(int adres1,int adres2,char value);
void TXString( char* string, int length );

void main(void){

    WDTCTL = WDTPW + WDTHOLD;                // Se para el Watchdog Ti-
mer

    // Bucle de inicialización necesario para el correcto funcionamiento
    del sistema
    {
        volatile int i;
        for(i = 0; i < 0xFFFF; i++){
        }

    ////////////////////////////////////INICIALIZACIÓN////////////////////////////////////
    ////////////////////////////////////

    //Inicialización de variables

    int volt=0;
    int posicion;
    int a,b,c, x, y, z;
    char datos[] = {"\r\nSURF00.SDAT"};
    // String que se enviará, donde 00 es el identificador de la superfi-
    cie y
    // DAT la posición del sistema
    datos[6]= '0' + ((ident/10)%10);
    datos[7]= '0' + (ident%10);

    posmax=posmax*10;    // Operación para la correcta ejecución del sis-
    tema

    //Inicialización de los puertos

    P3SEL |= 0x3C;                // P3.4,5 = USCI_A0 TXD/RXD
    P3DIR |= 0x03;
    P2DIR |= 0x00;
```

```
P4DIR |= 0x60;
P4OUT |= 0x60;

// Configuración SERIE (UART_A)

BCSCTL1 = CALBC1_8MHZ;           // Set DCO
DCOCTL = CALDCO_8MHZ;

BCSCTL3 |= LFXT1S_2;             // LFXT1 = VLO
TACCTL0 = CCIE;                  // TACCR0 interrupción
habilitada
TACCR0 = 12000;                  // ~1 second
TACTL = TASSEL_1 + MC_1;         // ACLK, upmode

UCA0CTL1 = UCSSEL_2;             // SMCLK
UCA0BR0 = 0x41;                  // 9600 from 8Mhz
UCA0BR1 = 0x3;
UCA0MCTL = UCBRS_2;
UCA0CTL1 &= ~UCSWRST;            // **Initialize USCI state
machine**
IE2 |= UCA0RXIE;                 // Enable USCI_A0 RX inte-
rrupt
__enable_interrupt();

// Configuración ETHERNET (UART_B)

UCB0CTL0 |= UCMSB + UCMST + UCSYNC;           // 3-pin, 8-bit SPI
mstr, LSB 1st
UCB0CTL1 |= UCSSEL_2;                         // SMCLK
UCB0BR0 = 0x02;
UCB0BR1 = 0;
UCB0CTL1 &= ~UCSWRST;                         // **Initialize USCI state
machine**

w5100_init();

/////////////////////////FUNCIONAMIENTO          DEL          SISTE-
MA////////////////////////////////////

while(1){

//Función 1: Recepción de datos del sensor
//Se realiza la función 1 directamente con la función 2

//Función 2: Amplificación de la señal recibida

OA0CTL1 = OAFBR_0 + OAFRC_4;    // Modo Amplificador No-Inversor, G=1
OA0CTL0 = OAPM_1+ OAN_1 + OAP_0 + OAADC1 ;
                                // Señal + del sensor conectada a OA0IO
(Pin 3)

                                // Salida conectada en el A1
```



```
//Función 3: Conversión A/D

ADC10AE0 |= 0x02; //Entrada por el puerto
A1
ADC10CTL1 = INCH_1 + ADC10DIV_4; // Entrada A1
ADC10CTL0 = SREF_1 + ADC10SHT_3 + ADC10ON + ADC10IE + ADC10SR + REFON
+ REF2_5V;
// Tensión máxima 2.5 V, mínima 0 V
for( int degC = 240; degC > 0; degC-- );//Retraso para fijar la con-
figuración
ADC10CTL0 |= ENC + ADC10SC; // Comienzo de la conversión
__bis_SR_register(CPUOFF + GIE); // LPM0 con interrupciones
habilitadas
volt=ADC10MEM; //guardamos en la variable VOLT el dato de la
conversión

ADC10CTL0 &= ~ENC;
ADC10CTL0 &= ~(REFON + ADC10ON); // se desactiva el A/D para ahorrar
energía

//Función 4: Tratamiento de Datos

// Ecuaciones para calcular la posición del sistema

a=(nmax-nmin)/10;
b=(posmax-posmin)/a;
c=b*nmin/100;
x=volt/10;
y=x*b;
z=y/10;

//Posición de la superficie, se guarda en la variable "posicion"
posicion=z-c+posmin+poffset;

if (b>a){ // Si b>a, ha habido un error en el cálculo de la posición
    datos[10]='E';
    datos[11]='R';
    datos[12]='R';}
else if(posicion>=0){
    datos[9]= '+';
    datos[10]='0' + ((posicion/100)%10);
    datos[11]='0' + ((posicion/10)%10);
    datos[12]='0' + (posicion)%10;}
else if(posicion<0){
    datos[9]= '-';
    datos[10]='0' + ((posicion/100)%10);
    datos[11]='0' + ((posicion/10)%10);
    datos[12]='0' + (posicion)%10;}
else{
    datos[10]='E';
```



```
datos[11]='R';
datos[12]='R';}

TXString (datos,sizeof datos);
w5100_udp (datos);

}
}

/////////////////////////FUNCIONES                                DEL                PROGRA-
MA/////////////////////////

// Se envía la información por Serie
void TXString( char* string, int length )
{
    int pointer;
    for( pointer = 0; pointer < length; pointer++)
    {
        volatile int i;
        UCA0TXBUF = string[pointer];
        while (!(IFG2&UCA0TXIFG));                // USCI_A0 TX buffer ready?
    }
}

//Interrupcion del ADC10
#pragma vector=ADC10_VECTOR
__interrupt void ADC10_ISR(void)
{
    __bic_SR_register_on_exit(CPUOFF);            // Clear CPUOFF bit from
0(SR)
}

/////////////////////////FUNCIONES                                DE                LA                CONEXIÓN                ETHER-
NET/////////////////////////

void w5100_init(void){
    for (int i=0;i<50;i++){
        P4OUT |= 0x40;
    }
    P4OUT |= 0x60;

    //Set gateway IP adress
    w5100_write(w5100_confad,w5100_gar0 , GIP0);
    w5100_write(w5100_confad,w5100_gar1 , GIP1);
    w5100_write(w5100_confad,w5100_gar2 , GIP2);
    w5100_write(w5100_confad,w5100_gar3 , GIP3);
    //Set Subnetmask
    w5100_write(w5100_confad,w5100_subr0 , DNS0);
    w5100_write(w5100_confad,w5100_subr1 , DNS1);
    w5100_write(w5100_confad,w5100_subr2 , DNS2);
    w5100_write(w5100_confad,w5100_subr3 , DNS3);
    //Set MAC
```



```
w5100_write(w5100_confad,w5100_shar0 , MAC0);
w5100_write(w5100_confad,w5100_shar1 , MAC1);
w5100_write(w5100_confad,w5100_shar2 , MAC2);
w5100_write(w5100_confad,w5100_shar3 , MAC3);
w5100_write(w5100_confad,w5100_shar4 , MAC4);
w5100_write(w5100_confad,w5100_shar5 , MAC5);
//Set own IP address
w5100_write(w5100_confad,w5100_sipr0 , IP0);
w5100_write(w5100_confad,w5100_sipr1 , IP1);
w5100_write(w5100_confad,w5100_sipr2 , IP2);
w5100_write(w5100_confad,w5100_sipr3 , IP3);

}

void w5100_write(int adres1,int adres2,int value){

    while (!(IFG2 & UCB0TXIFG));           // USCI_A0 TX buffer ready?
    UCB0TXBUF = 0xF0;                       // Byte to SPI TXBUF
    for(int i = 0xEFFF; i > 0; i--);        // Delay
    while (!(IFG2 & UCB0TXIFG));
    UCB0TXBUF = adres1;
    for(int i = 0xEFFF; i > 0; i--);
    while (!(IFG2 & UCB0TXIFG));
    UCB0TXBUF = adres2;
    for(int i = 0xEFFF; i > 0; i--);
    while (!(IFG2 & UCB0TXIFG)); //
    UCB0TXBUF = value;
    for(int i = 0xEFFF; i > 0; i--);

}

void w5100_data(int adres1,int adres2,char value){

    while (!(IFG2 & UCB0TXIFG));           // USCI_A0 TX buffer ready?
    UCB0TXBUF = 0xF0;                       // Byte to SPI TXBUF
    for(int i = 0xEFFF; i > 0; i--);        // Delay
    while (!(IFG2 & UCB0TXIFG));
    UCB0TXBUF = adres1;
    for(int i = 0xEFFF; i > 0; i--);
    while (!(IFG2 & UCB0TXIFG));
    UCB0TXBUF = adres2;
    for(int i = 0xEFFF; i > 0; i--);
    while (!(IFG2 & UCB0TXIFG));
    UCB0TXBUF = value;
    for(int i = 0xEFFF; i > 0; i--);

}

int w5100_read(int adres1,int adres2){
    int data;
```



```
while (!(IFG2 & UCB0TXIFG));          // USCI_A0 TX buffer ready?
UCB0TXBUF = 0x0F;                      // Byte to SPI TXBUF
for(int i = 0xEFFF; i > 0; i--);      // Delay
while (!(IFG2 & UCB0TXIFG));
UCB0TXBUF = adres1;
for(int i = 0xEFFF; i > 0; i--);
while (!(IFG2 & UCB0TXIFG));
UCB0TXBUF = adres2;
while (!(IFG2 & UCB0RXIFG));
data = UCB0RXBUF;
return data;
}

void w5100_udp(char* data)
{
    //Se abre el Socket 0 como UDP

    // Se asigna la misma memoria a todos los sockets para transmisión y
    recepción
    w5100_write(w5100_confad, w5100_rmsr , 0x55);
    w5100_write(w5100_confad, w5100_tmsr , 0x55);

    //Se asigna el protocolo UDP
    w5100_write(w5100_s0_conf, w5100_s0_mr , Sn_mr_udp);

    //Port 5000
    w5100_write(w5100_s0_conf, w5100_s0_port0 , iport0);
    w5100_write(w5100_s0_conf, w5100_s0_port1 , iport1);

    //Se manda abrir el puerto
    w5100_write(w5100_s0_conf, w5100_s0_cr , Sn_cr_open);

    //Se espera a que se haya abierto el socket
    int value=1;
    while(value!=Sock_udp) {
        value = w5100_read(w5100_s0_conf, w5100_s0_sr);
    }

    // Se envía a una IP o mediante BROADCAST la información
    // El sistema está configurado para realizar un BROADCAST

    w5100_write(w5100_s0_conf, w5100_s0_dipr0 , IPD0);
    w5100_write(w5100_s0_conf, w5100_s0_dipr1 , IPD1);
    w5100_write(w5100_s0_conf, w5100_s0_dipr2 , IPD2);
    w5100_write(w5100_s0_conf, w5100_s0_dipr3 , IPD3);

    // Puerto de destino del receptor

    w5100_write(w5100_s0_conf, w5100_s0_dport0 , dport0);
    w5100_write(w5100_s0_conf, w5100_s0_dport1 , dport1);

    //Envío de la información de la posición del sistema al módulo de
    ethernet
    w5100_data(0x40,0x00 , data[0]);
}
```

```
w5100_data(0x40,0x01 , data[1]);
w5100_data(0x40,0x02 , data[2]);
w5100_data(0x40,0x03 , data[3]);
w5100_data(0x40,0x04 , data[4]);
w5100_data(0x40,0x05 , data[5]);
w5100_data(0x40,0x06 , data[6]);
w5100_data(0x40,0x07 , data[7]);
w5100_data(0x40,0x08 , data[8]);
w5100_data(0x40,0x09 , data[9]);
w5100_data(0x40,0x0A , data[10]);
w5100_data(0x40,0x0B , data[11]);
w5100_data(0x40,0x0C , data[12]);

//Envío del tamaño de la información enviada (13 bytes)
w5100_write(w5100_s0_conf, w5100_s0_tx_wr0 , 0x00);
w5100_write(w5100_s0_conf, w5100_s0_tx_wr1 , 0x0D);

//Envío de la información a través de la red
w5100_write(w5100_s0_conf, w5100_s0_cr , Sn_cr_send);

//Se realiza un bucle hasta que el registro de órdenes queda limpio
value=1;
while(value!=0){
    value = w5100_read(w5100_s0_conf, w5100_s0_cr);
}

//Interrupción del socket 0
w5100_write(w5100_confad,w5100_imr , 0x01);
}
```



## 2 SISTEMA.H

```
// Esta es la ÚNICA hoja donde se tienen que modificar las variables
según

// el sistema que se vaya a utilizar. Existe otra hoja con variables a
modificar

// para la conexión ETHERNET (ethernet.h) Mirar el manual de
instalación para

// información más detallada

/* INFORMACIÓN DEL SISTEMA A MEDIR */

//Posición mínima de la superficie
int posmin = 0 ;

//Posición máxima de la superficie (mayor o igual a 0)
int posmax = 190 ;

//Tensión mínima en formato digital*
int nmin = 60 ;

//Tensión máxima en formato digital*
int nmax = 810 ;

//Identificador numérico de la superficie

int ident = 00;

/*Para más información sobre cómo obtener este dato dirigirse al
manual
```



### 3 ETHERNET.H

```
// Esta es la ÚNICA hoja donde se tienen que modificar las variables  
para  
// la conexión Ethernet. Existe otra hoja con variables a modificar  
// para cada sistema (sistema.h) . Mirar el manual de instalación para  
// información más detallada
```

```
////////////////////DATOS      DE      CONFIGURACIÓN      DEL      SISTE-  
MA////////////////////
```

```
// Puerta de enlace del sistema (GIP0.GIP1.GIP2.GIP3)
```

```
int GIP0 = 172;  
int GIP1 = 26;  
int GIP2 = 0;  
int GIP3 = 1;
```

```
// DNS submask del sistema (DNS0.DNS1.DNS2.DNS3)
```

```
int DNS0 = 255;  
int DNS1 = 255;  
int DNS2 = 255;  
int DNS3 = 0;
```

```
// Dirección física (MAC) del sistema (MAC0.MAC1.MAC2.MAC3.MAC4.MAC5);
```

```
int MAC0 = 0xAA;  
int MAC1 = 0x10;  
int MAC2 = 0x20;  
int MAC3 = 0x30;  
int MAC4 = 0x40;  
int MAC5 = 0x50;
```

```
// Dirección IP del sistema (IP0.IP1.IP2.IP3)
```

```
int IP0 = 172;  
int IP1 = 26;  
int IP2 = 0;  
int IP3 = 111;
```

```
// Puerto del socket abierto del sistema (en hexadecimal)  
// Mirar en el manual para más indicaciones  
// Puerto configurado: 5000
```

```
int iport0 = 0x13;  
int iport1 = 0x88;
```



Diseño de  
Sistema Sensor  
Para superficies  
de UAV

Iss./Rev.: D01  
Date: 07/07/2009

---

```
/////////////////////////////////DATOS DEL SISTEMA RECEPTOR/////////////////////////////////
```

```
// IP a la que se envía la información      (IPD0.IPD1.IPD2.IPD3)  
// Configurado en modo BROADCAST
```

```
int IPD0 = 255;  
int IPD1 = 255;  
int IPD2 = 255;  
int IPD3 = 255;
```

```
// Puerto al que se envía la información (en hexadecimal)  
// Mirar en el manual para más indicaciones  
// Puerto configurado: 4369
```

```
int dport0 = 0x11;  
int dport1 = 0x11;
```





## 4 W5100REG.H

//Registros utilizados en la programación del módulo WIZ810MJ (W5100)

//Common registers

//Mode register

int w5100\_mr = 0x0000;

int w5100\_confad = 0x00;

//Gateway address registers

int w5100\_gar0 = 0x01;

int w5100\_gar1 = 0x02;

int w5100\_gar2 = 0x03;

int w5100\_gar3 = 0x04;

//Subnet mask Address registers

int w5100\_subr0 = 0x05;

int w5100\_subr1 = 0x06;

int w5100\_subr2 = 0x07;

int w5100\_subr3 = 0x08;

//Source Hardware Address registers

int w5100\_shar0 = 0x09;

int w5100\_shar1 = 0x0A;

int w5100\_shar2 = 0x0B;

int w5100\_shar3 = 0x0C;

int w5100\_shar4 = 0x0D;

int w5100\_shar5 = 0x0E;

//Source IP Address registers

int w5100\_sipr0 = 0x0F;

int w5100\_sipr1 = 0x10;

int w5100\_sipr2 = 0x11;

int w5100\_sipr3 = 0x12;

//reserved

//0x0013

//0x0014

//interrupt

int w5100\_ir = 0x15;

int w5100\_imr = 0x16;

//retry time

int w5100\_rtr0 = 0x17;

int w5100\_rtr1 = 0x18;

```
int w5100_rcr = 0x19;
//memory
int w5100_rmsr = 0x1A;
int w5100_tmsr = 0x1B;
//PPoe
int w5100_patr0 = 0x1C;
int w5100_patr1 = 0x1D;
//reserved
//0x001e
//0x001f
//0x0020
//0x0021
//0x0022
//0x0023
//0x0024
//0x0025
//0x0026
//0x0027
//request timer
int w5100_ptimer = 0x28;
int w5100_pmagic = 0x29;
//unreachable IP-address
int w5100_uipr0 = 0x2A;
int w5100_uipr1 = 0x2B;
int w5100_uipr2 = 0x2C;
int w5100_uipr3 = 0x2D;
//unreachable port
int w5100_uport0 = 0x2E;
int w5100_uport1 = 0x2F;
//reserved
//0x0030 - 0x03FF
//
//
//Socket 0 registers

int w5100_s0_conf = 0x04;
int w5100_s0_mr = 0x00;
int w5100_s0_cr = 0x01;
int w5100_s0_ir = 0x02;
int w5100_s0_sr = 0x03;
//Socket 0 Source Port
int w5100_s0_port0 = 0x04;
int w5100_s0_port1 = 0x05;
//Socket 0 Destination Hardware Address
int w5100_s0_dhar0 = 0x06;
int w5100_s0_dhar1 = 0x07;
int w5100_s0_dhar2 = 0x08;
int w5100_s0_dhar3 = 0x09;
int w5100_s0_dhar4 = 0x0A;
int w5100_s0_dhar5 = 0x0B;
//Socket 0 Destination IP Address
int w5100_s0_dipr0 = 0x0C;
int w5100_s0_dipr1 = 0x0D;
int w5100_s0_dipr2 = 0x0E;
```

```
int w5100_s0_dipr3 = 0x0F;
//Socket 0 Destination Port
int w5100_s0_dport0 = 0x10;
int w5100_s0_dport1 = 0x11;
//Socket 0 Maximum Segment Size
int w5100_s0_mssr0 = 0x12;
int w5100_s0_mssr1 = 0x13;
//Socket 0 Protocol in IP Raw mode
int w5100_s0_proto = 0x14;
int w5100_s0_tos = 0x15;
int w5100_s0_ttl = 0x16;
//reserved
//0x0417 - 0x041F
//Socket 0 TX Free size
int w5100_s0_tx_fsr0 = 0x20;
int w5100_s0_tx_fsr1 = 0x21;
//Socket 0 TX Read Pointer
int w5100_s0_tx_rd0 = 0x22;
int w5100_s0_tx_rd1 = 0x23;
//Socket 0 TX write Pointer
int w5100_s0_tx_wr0 = 0x24;
int w5100_s0_tx_wr1 = 0x25;
//Socket 0 RX Received Size
int w5100_s0_rx_rsr0 = 0x26;
int w5100_s0_rx_rsr1 = 0x27;
//Socket 0 RX Read Pointer
int w5100_s0_rx_rd0 = 0x28;
int w5100_s0_rx_rd1 = 0x29;
//reserved
//0x042A - 0x04FF
//
//

// Sn_mr values
int Sn_mr_close = 0x00; // unused socket
int Sn_mr_tcp = 0x01; // TCP
int Sn_mr_udp = 0x02; // UDP
int Sn_mr_ipraw = 0x03; // IP LAYER RAW SOCK
int Sn_mr_macraw = 0x04; // MAC LAYER RAW SOCK
int Sn_mr_pppoe = 0x05; // PPPoE
int Sn_mr_nd = 0x20; // No Delayed Ack(TCP) flag
int Sn_mr_multi = 0x80; // support multicating

//Sn_cr values
int Sn_cr_open = 0x01; // initialize or open socket
int Sn_cr_listen = 0x02; // wait connection
request in tcp mode(Server mode)
int Sn_cr_connect = 0x04; // send connection re-
quest in tcp mode(Client mode)
int Sn_cr_discon = 0x08; // send closing regeuset in tcp mode
int Sn_cr_close = 0x10; // close socket
int Sn_cr_send = 0x20; // updata txbuf pointer, send data
int Sn_cr_send_mac = 0x21; // send data with MAC ad-
dress, so without ARP process
```



```
int Sn_cr_send_keep = 0x22;           // send keep alive message
int Sn_cr_recv = 0x40;                // Update Rxbuf Pointer , Recv Data

//Sn_sr values
int Sock_closed = 0x00;                // closed
int Sock_init = 0x13;                 // init state
int Sock_listen = 0x14;               // listen state
int Sock_syntent = 0x15;              // Connection State
int Sock_synrecv = 0x16;             // connection state
int Sock_established = 0x17;         // success to connect
int Sock_fin_wait = 0x18;            // closing state
int Sock_closing = 0x1A;             // closing state
int Sock_time_wait = 0x1B;           // closing state
int Sock_close_wait = 0x1C;          // closing state
int Sock_last_ack = 0x1D;            // closing state
int Sock_udp = 0x22;                 // udp socket
int Sock_ipraw = 0x32;               // ip raw mode socket
int Sock_macraw = 0x42;              // mac raw mode socket
int Sock_pppoe = 0x5F;               // Pppoe Socket Er , Recv Data
```

## 5 CALIBRACION.C

```
#include "msp430x22x4.h"

void TXString( char* string, int length );

void main(void){

    WDTCTL = WDTPW + WDTHOLD;                // Se para el Watchdog Timer

    // Bucle de inicialización necesario para el correcto funcionamiento
    // del sistema
    {
        volatile int i;
        for(i = 0; i < 0xFFFF; i++){
        }
    }

    //Inicialización de variables

    int volt=0;
    char  datos[] = {"\r\nVOLT.DATA"};

    //Inicialización de los puertos

    P3SEL |= 0x3C;                            // P3.4,5 = USCI_A0 TXD/RXD
    P3DIR |= 0x03;
    P2DIR |= 0x00;

    // Configuración SERIE (UART_A)

    BCSCCTL1 = CALBC1_8MHZ;                    // Set DCO
    DCOCTL = CALDCO_8MHZ;

    BCSCCTL3 |= LFXLT1S_2;                    // LFXLT1 = VLO
    TACCTL0 = CCIE;                            // TACCR0 interrupción
    // TACCR0 interrupción
    TACCR0 = 12000;                            // ~1 second
    TACTL = TASSEL_1 + MC_1;                    // ACLK, upmode

    UCA0CTL1 = UCSSEL_2;                       // SMCLK
    UCA0BR0 = 0x41;                            // 9600 from 8Mhz
    UCA0BR1 = 0x3;
    UCA0MCTL = UCBRS_2;
    UCA0CTL1 &= ~UCSWRST;                      // **Initialize USCI state
    machine**
    IE2 |= UCA0RXIE;                           // Enable USCI_A0 RX interrupt
    __enable_interrupt();
```

```
while(1){

    OA0CTL1 = OAFBR_0 + OAFC_4;    // Modo Amplificador No-Inversor, G=1
    OA0CTL0 = OAPM_1+ OAN_1 + OAP_0 + OAADC1 ;
                                   // Señal + del sensor conectada a OA0IO
(Pin 3)
                                   // Salida conectada en el A1

    //Función 3: Conversión A/D

    ADC10AE0 |= 0x02;                //Entrada por el puerto
A1
    ADC10CTL1 = INCH_1 + ADC10DIV_4;    // Entrada A1
    ADC10CTL0 = SREF_1 + ADC10SHT_3 + ADC10ON + ADC10IE + ADC10SR + REFON
+ REF2_5V;
    // Tensión máxima 2.5 V, mínima 0 V
    for( int degC = 240; degC > 0; degC-- );//Retraso para fijar la con-
figuración
    ADC10CTL0 |= ENC + ADC10SC;        // Comienzo de la conversión
    __bis_SR_register(CPUOFF + GIE);    // LPM0 con interrupciones
habilitadas
    volt=ADC10MEM;                    //guardamos en la variable VOLT el dato de la
conversión

    ADC10CTL0 &= ~ENC;
    ADC10CTL0 &= ~(REFON + ADC10ON);    // se desactiva el A/D para ahorrar
energia

    datos[7]='0'+((volt/1000)%10);
    datos[8]='0'+((volt/100)%10);
    datos[9]='0'+((volt/10)%10);
    datos[10]='0'+(volt)%10;

    TXString(datos, sizeof datos);

}
}
```



```
//////////FUNCIONES DEL PROGRAMA//////////

// Se envía la información por Serie
void TXString( char* string, int length )
{
    int pointer;
    for( pointer = 0; pointer < length; pointer++)
    {
        volatile int i;
        UCA0TXBUF = string[pointer];
        while (!(IFG2&UCA0TXIFG));           // USCI_A0 TX buffer ready?
    }
}

//Interrupcion del ADC10
#pragma vector=ADC10_VECTOR
__interrupt void ADC10_ISR(void)
{
    __bic_SR_register_on_exit(CPUOFF);       // Clear CPUOFF bit from
0(SR)
}
```



Diseño de  
Sistema Sensor  
Para superficies  
de UAV

Iss./Rev.: D01  
Date: 07/07/2009

---

[END OF DOCUMENT]